目　录

光盘目录

第1章　辨识的基本概念

在社会和生产中，越来越多需要辨识过程（或系统）模型的问题已广泛引起人们的重视，社会科学和自然科学的各个领域有很多学者在研究有关线性和非线性的辨识问题。对于线性系统的模型辨识和参数估计，早在20世纪60年代初期，Zadeh就给出了系统辨识的定义，人们已经进行了深入的研究，并总结出一套成熟的方法：最小二乘辨识方法、最大似然辨识方法、梯度法辨识等等。这些理论和方法已在工程实际中得到了广泛的应用。然而在现实中，非线性是普遍存在的，而线性模型只是对非线性的一种简化和近似。对非线性系统的研究、设计要比线性系统复杂得多。且方法并非惟一，更找不到统一的设计模式。只能是针对具体问题分析其非线性的问题所在，抓住其影响系统静、动态品质的要害，研究辨识非线性系统模型及控制的理论和方法，进而对系统进行辨识、补偿或控制。若能够通过辨识得到其较准确的模型，则是控制问题的关键。本章主要介绍系统辨识的基本概念：1.1节系统和模型，其中包括模型的表现形式及数学模型的分类；1.2节辨识的定义；1.3节辨识问题的表示形式及原理，其中包括辨识问题的表达形式、辨识算法的基本原理和误差准则；1.4节辨识的内容和步骤；1.5节介绍典型的非线性系统辨识与控制方法。

1.1　系统和模型

1.1.1　模型的表现形式

系统是通过模型来表达的，因此系统辨识也称为模型辨识。模型有如下一些表现形式：

（1）“直觉”模型。它指过程的特性以非解析的形式直接储存在人脑中，靠人的直觉控制过程的进行。例如，司机就是靠“直觉模型”来控制汽车的方向盘。

（2）物理模型。它是根据相似原理把时间过程加以缩小的复制品，或是实际过程的一种物理模拟。比如：电力系统动态模型、某种控制机床模型或风洞、水利学模型、传热学模型等均是物理模型。

（3）图表模型。它以图形或表格的形式来表现过程的特性。如阶跃响应、脉冲响应和频率响应等，也称非参数模型。

（4）数学模型。它用数学结构的形式来反映实际过程的行为特性。常用的有代数方程、微分方程、差分方程和状态方程。

a. 经济学上的 Cobb-Douglas 产生关系模型(代数方程)

$$Y = AL^{a_1}K^{a_2}, \quad a_1 > 0; \quad a_2 < 1 \tag{1.1}$$

式中,Y 为产值;L 为劳动力;K 为资本。

b. 微分方程

$$\begin{aligned} & z^{(n)}(t) + a_1 z^{(n-1)}(t) + a_2 z^{(n-2)} \cdots + a_{n-1} z^{(1)}(t) + a_n z(t) \\ & = b_1 u^{(m-1)}(t) + b_2 u^{(m-2)} \cdots + a_{m-1} u^{(1)}(t) + b_m u(t) + e(t) \end{aligned} \tag{1.2}$$

式中,$u(t)$和 $z(t)$为输入输出量;$e(t)$为噪声项。

c. 差分方程

$$A(z^{-1})z(k) = B(z^{-1})u(k) + e(k) \tag{1.3}$$

式中

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots a_{n_a} z^{n_a} \\ B(z^{-1}) &= b_1 z^{-1} + b_2 z^{-2} + \cdots b_{n_b} z^{n_b} \end{aligned} \tag{1.4}$$

即

$$\begin{aligned} & z(k) - a_1 z(k-1) + a_2 z(k-2) + \cdots + a_{n_a} z(k - n_a) \\ & = b_1 u(k-1) + \cdots + b_{n_b} u(k - n_b) + e(k) \end{aligned} \tag{1.5}$$

式中,$u(k)$和 $z(k)$为输入输出量;$e(k)$为噪声项;z^{-1}表示延迟算子,即 $z^{-1}x(k) = x(k-1)$。

d. 状态方程

$$\begin{aligned} \dot{x}(t) &= Ax(t) + bu(t) + F\omega(t) \\ z(t) &= cx(t) + hw(t) \end{aligned} \tag{1.6}$$

或

$$\begin{aligned} x(k+1) &= Ax(k) + bu(k) + F\omega(k) \\ z(k) &= cx(k) + hw(k) \end{aligned} \tag{1.7}$$

式中,$u(\cdot)$和 $z(\cdot)$为输入输出量;$x(\cdot)$为状态变量;$\omega(\cdot)$和 $w(\cdot)$为噪声项。

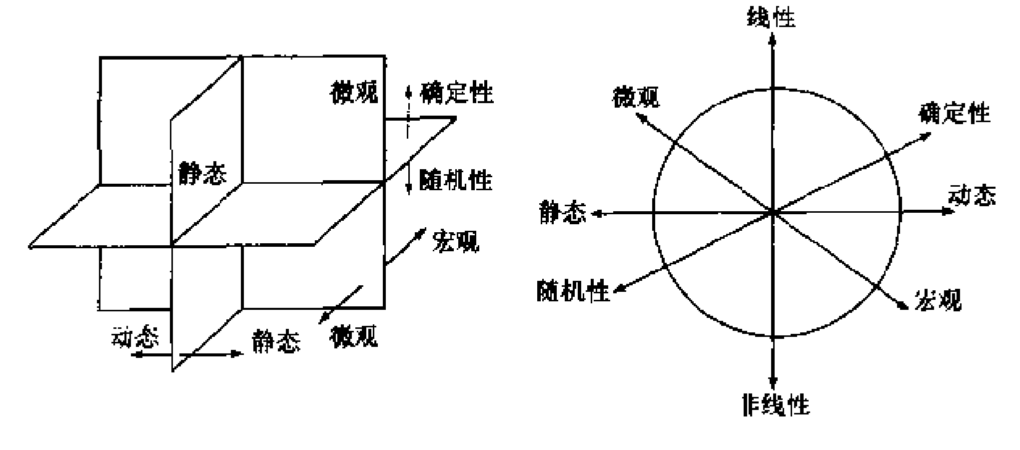


图 1.1 数学模型的分类

1.1.2 数学模型的分类

一般来说，系统的特性有线性与非线性、动态与静态、确定性与随机性、宏观与微观之分，故描述系统特性的数学模型必然也有这几种类型的区分。这些类型的数学模型可以用图 1.1 来表示。另外，也有参数模型和非参数模型；外部模型(SISO 系统)和内部模型(MIMO 系统)等不同分类。

1.2 辨识建模的定义

系统模型建立(建模)分为机理建模、系统辨识建模、机理分析和系统辨识相结合的建模方法。①机理建模是一种常用的建模方法，是根据系统的结构，分析系统运动的规律，利用已知的相应的定律、定理或原理，如化学动力学原理、生物学定律、牛顿定理、能量平衡方程和传热传质原理等推导出描述系统的数学模型，建立的模型可能是线性的或非线性的，这类建模有时也称为白箱建模。②系统辨识是一种利用系统的输入输出数据建模的方法，是黑箱建模问题，即使对系统的结构和参数一无所知，也可以通过多次测量得到的系统的输入和输出的数据来求得系统的模型，是对实际系统的一个合适的近似。在这方面线性系统的建模(辨识)理论已成熟，有关学科的专业知识要求不变，获得的模型较简单。③机理建模和辨识建模结合的方法适用于系统的运动机理不是完全未知的情况，称之为灰色建模。利用已知的运动机理和经验确定系统的结构和参数(即确定模型)。

辨识问题包括模型结构辨识和参数估计。所谓参数估计或点估计问题，即设 x 为一未知参数，可以视为参数空间 X 中的一个点，测量 $\boldsymbol{y}$ 是一随机向量，其分量依赖于参数 x。即根据 $\boldsymbol{y}$ 的一组样本(观测值)对参数 x 的估计就称为参数估计问题。而系统辨识是研究如何获得必要的系统输入输出的数据(样本)，以及如何从所获得的数据构造一个相对真实地反映客观对象的数学模型。L. A. Zadeh 在 1962 年曾给系统辨识下过一个定义：

定义 1.1 辨识就是在输入和输出的基础上由规定的一类系统模型中确定一个系统模型，使之与被测系统等价。

这个定义明确了辨识的三大要素：系统的输入输出数据、模型类和等价准则。这个定义中提到的“一类系统模型”是指规定的连续时间模型或离散时间模型、输入输出模型或状态空间模型、确定性模型或随机模型、线性模型或非线性模型等。模型类的规定是根据人们对实际系统的了解以及建立模型的目的设定的。规定了模型类后，再由输入输出数据按结构辨识的方法确定系统的结构参数，并且用参数辨识的方法辨识系统的参数。

根据定义，我们所建立的模型必须与被测系统在某种意义上是等价的。

设 M 表示被测系统的一个模型，并且满足

$$M \in G_m \tag{1.8}$$

G_m 是具有某种属性的模型类。一个系统可看做是从系统允许的输入空间 U 到输出空间 Y 的一个算子 p（相当于对象 M），p 属于某个算子类 G。系统辨识的问题是指对于一个给定的算子类 G 和一个对象 $p \in G$，确定一个模型类 G_m 及它的一个元素 $p_m \in G_m$（可以认为 G_m 是 G 的一个子集），使得 p_m 尽可能的逼近 p。在实际的被控系统中，可以采集到对象的输入和输出的数据，现认为时间函数 $u(t)$，$y(t)$，$t \in [0, T]$ 所表示的输入、输出对可以定义这个系统 p，这时辨识的目标是确定 p_m，并且满足

$$\| y(u) - y_m(u) \| = \| p(u) - p_m(u) \| \leqslant \varepsilon, u \in U, \varepsilon > 0 \tag{1.9}$$

模型不确定的非线性系统的辨识应属于黑箱辨识问题。对于黑箱辨识方法，被测对象所属的算子是未知的，由于输入、输出对隐含地表示算子 p，但通常可认为 p 属于非线性连续算子集合，因此可选用对连续算子集合具有任意逼近能力的模型集合。对模型集合的选择主要考虑算法的简单性、模型的适应能力和逼近的精度等因素。对于离散时间系统，这样的集合有基于多项式的 NARMA 模型，而基于神经网络所构成的模型也属其中的一种。黑箱法由如下三个步骤组成：实验观测、数学建模、模型验证。

1.3　辨识问题的表示形式及原理

1.3.1　辨识问题的表达形式

下面着重讨论线性离散模型的辨识问题。所谓线性离散模型是指一个或几个变量可以表示的另外一些变量在时间或空间的离散点上的线性组合，如图 1.2 所示。

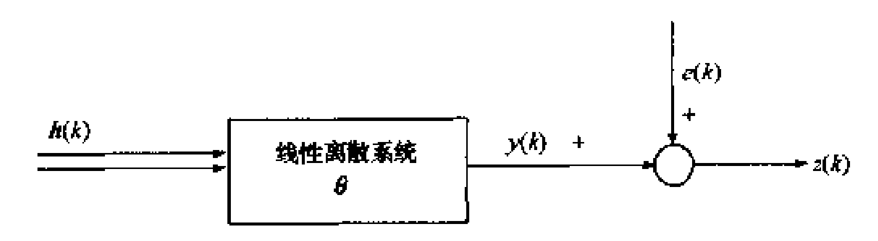


图 1.2　线性离散模型的数学表达形式

图中，$\boldsymbol{h}(k)$ 和 $z(k)$ 是模型的输入输出变量，它们在离散点上必须是可观测的；$e(k)$ 是模型噪声；$\boldsymbol{\theta}$ 是未知模型参数。记

$$\begin{cases} \boldsymbol{h}(k) = [h_1(k), h_2(k), h_3(k), \cdots, h_n(k)]^{\mathrm{T}} \\ \boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \cdots, \theta_N]^{\mathrm{T}} \end{cases} \tag{1.10}$$

则线性离散模型的输出可表示成

$$z(k) = \sum_{i=1}^{N} \theta_i h_i(k) + e(k) = \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta} + e(k) \tag{1.11}$$

例 1.1　将差分方程化成最小二乘格式。

考虑如下差分方程

$$\begin{aligned} & z(k) + a_1 z(k-1) + a_2 z(k-2) + \cdots + a_n z(k-n) \\ = & b_1 u(k-1) + \cdots + b_n u(k-n) + e(k) \end{aligned} \tag{1.12}$$

式中，方程的输入输出变量 $u(\cdot)$、$z(\cdot)$在各离散点上都是可观测的。

解　设样本及参数集为

$$\begin{cases} \boldsymbol{h}(k) = [-z(k-1), -z(k-2), \cdots, -z(k-n), u(k-1), \\ \qquad u(k-2), \cdots, u(k-n)]^{\mathrm{T}} \\ \boldsymbol{\theta} = [a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n]^{\mathrm{T}} \end{cases} \tag{1.13}$$

则 $\boldsymbol{h}(k)$是可观测的向量，那么差分方程所对应的最小二乘格式为

$$z(k) = \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta} + e(k) \tag{1.14}$$

如果图 1.3 是被辨识的系统，则描述它的模型必须是能化成图 1.4 所示的辨识表达形式，即最小二乘格式，输出量 $z(k)$应是输入量 $\boldsymbol{h}(k)$ 的线性组合，如式(1.14)。

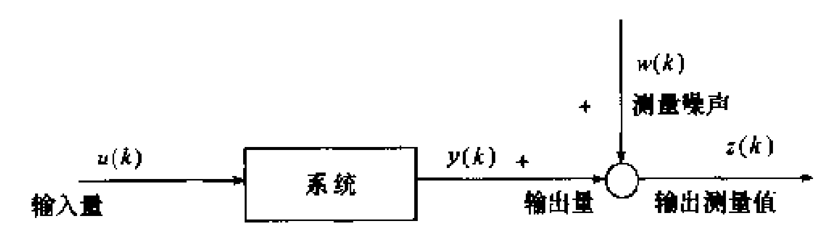


图 1.3　待辨识的过程

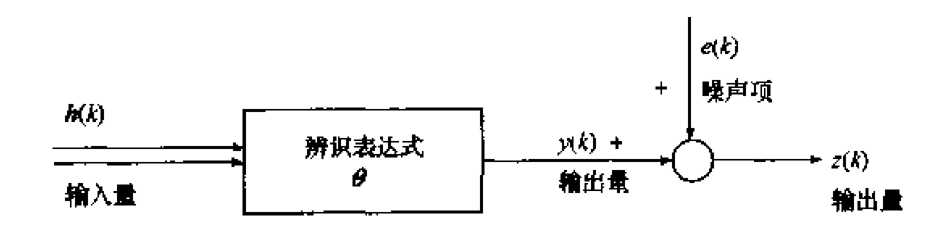


图 1.4　辨识问题的表达形式

1.3.2 辨识算法的基本原理

辨识的目的就是根据系统的测量信息,在某种准则意义下,估计出模型的未知参数,其基本原理如图 1.5 所示。

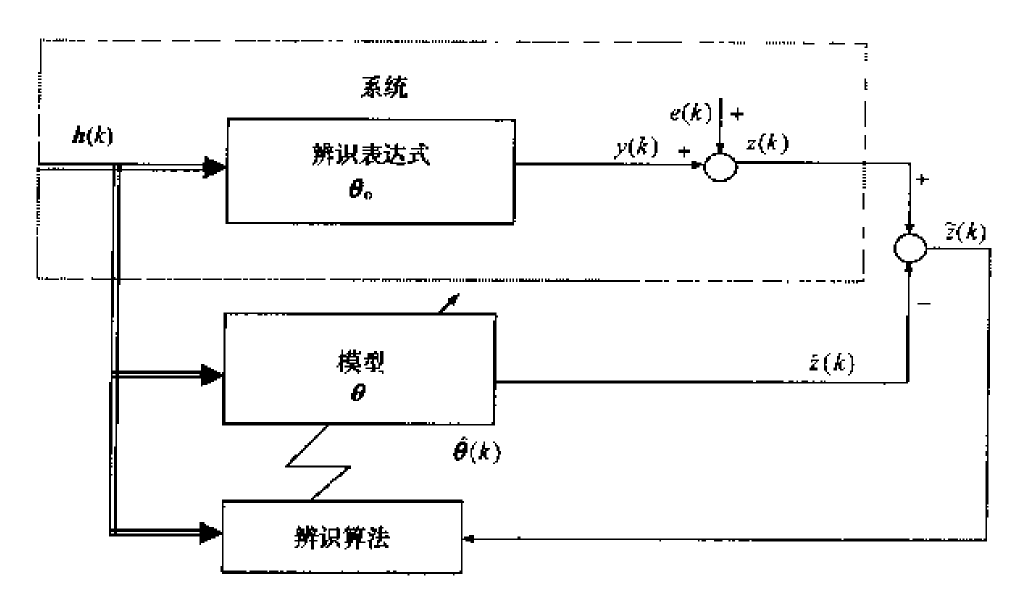


图 1.5　辨识原理

为了得到模型参数 θ 的估计值,通常采用逐步逼近的办法。在 k 时刻,根据前一时刻的估计参数计算出模型该时刻的输出,即系统输出预报值

$$\hat{z}(k)=\boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1) \tag{1.15}$$

同时计算出预报误差,或称新息(innovation)

$$\tilde{z}(k)=z(k)-\hat{z}(k) \tag{1.16}$$

图中,系统输出量

$$z(k)=\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}_0(k-1)+e(k) \tag{1.17}$$

及辨识表达式的输入量 $\boldsymbol{h}(k)$ 都是可测量的。然后将预报误差 $\tilde{z}(k)$ 反馈到辨识算法中,在某种准则下计算出 k 时刻的模型参数估计值 $\hat{\boldsymbol{\theta}}(k)$,并据此更新模型参数。这样依次迭代下去,直至其准则函数达到最小值。此刻模型的输出 $\hat{z}(k)$ 便可以在该准则意义下最好的逼近系统的输出 $z(k)$,从而获得了所需要的模型。

上述辨识算法原理可以推广到多输出系统。如果系统的输出是 m 维向量,那么辨识问题的表达形式应为

$$\boldsymbol{z}(k)=\boldsymbol{H}(k)\boldsymbol{\theta}+\boldsymbol{e}(k) \tag{1.18}$$

式中,输出向量为

$$\boldsymbol{z}(k)=[z_1(k),z_2(k),\cdots,z_m(k)]^{\mathrm{T}} \tag{1.19}$$

噪声向量为

$$e(k)=[e_1(k),e_2(k),\cdots,e_m(k)]^{\mathrm{T}} \tag{1.20}$$

参数向量为

$$\boldsymbol{\theta}=[\theta_1,\theta_2,\cdots,\theta_N]^{\mathrm{T}} \tag{1.21}$$

输入数据阵为

$$\boldsymbol{H}(k)=\begin{bmatrix} h_{11}(k) & h_{12}(k) & \cdots & h_{1N}(k) \\ h_{21}(k) & h_{22}(k) & \cdots & h_{2N}(k) \\ \vdots & \vdots & & \vdots \\ h_{m1}(k) & h_{m2}(k) & \cdots & h_{mN}(k) \end{bmatrix} \tag{1.22}$$

该多输出情况下的辨识问题与单输出情况下的辨识问题相同,其辨识原理如图1.6所示。

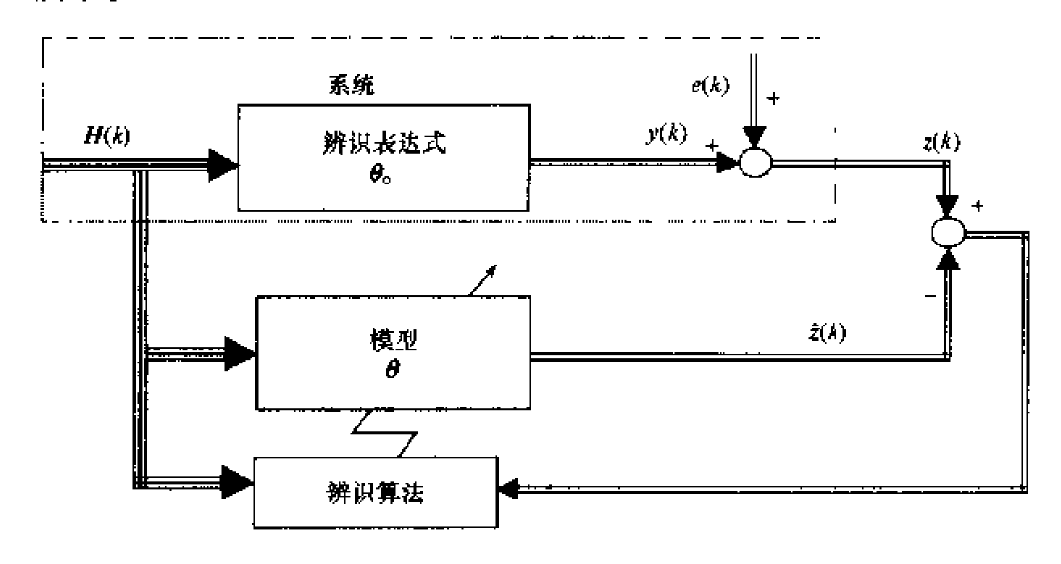


图1.6　多输出过程的辨识原理

1.3.3　误差准则

等价准则是辨识问题中不可缺少的三大要素之一,它是用来衡量模型接近实际过程的准则,而且它通常被表示为一个误差的泛函。因此等价准则也叫做误差准则或损失函数,也称准则函数,记作

$$J(\theta)=\sum_{k=1}^{L} f(\varepsilon(k)) \tag{1.23}$$

式中,$f(\cdot)$是$\varepsilon(k)$的函数。用得最多的是平方函数,即

$$f(\varepsilon(k))=\varepsilon^2(k) \tag{1.24}$$

$\varepsilon(k)$是定义在区间$(0,L)$上的误差函数。这个误差函数应该广义地理解为模型与实际过程的"误差",也可以是输出误差、输入误差或广义误差,如图1.7所示。

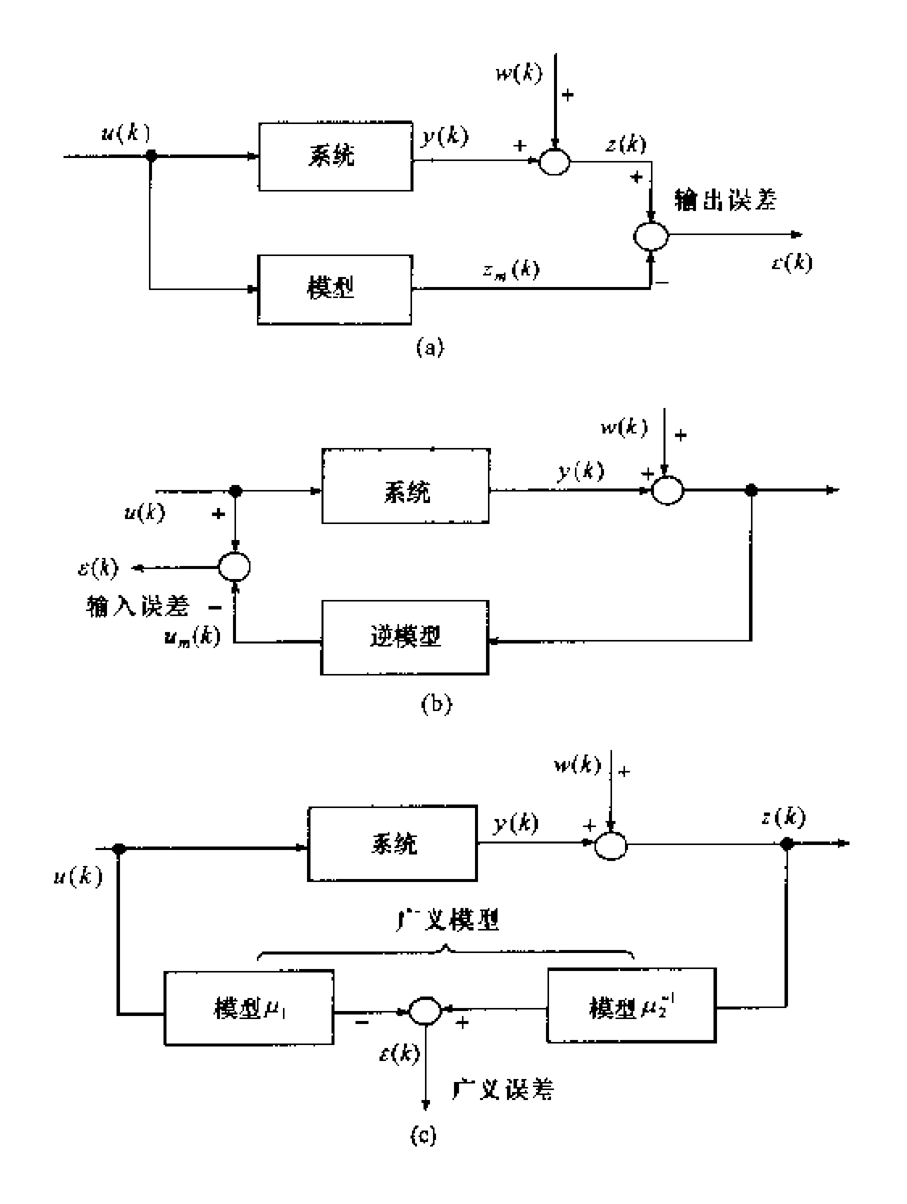


图 1.7　误差准则分类

从图 1.7 可知,将系统的输出和模型的输出的差定义的误差称为输出误差;将系统的输入和逆模型的输出的差定义的误差称为输入误差;而从系统广义模型 μ_i 定义的误差称作广义误差。输出误差在辨识中是应用最广泛的一种误差准则,如图 1.7(a)所示。当实际系统和模型的输出分别记作 $z(k)$和 $z_m(k)$时,则

$$\varepsilon(k) = z(k) - z_m(k) = z(k) - \mu[u(k)] \tag{1.25}$$

称作输出误差。式中,$\mu[u(k)]$ 是当输入为 $u(k)$ 时的模型输出。如果扰动是作用在过程输出端的白噪声,那么选用这种误差准则就是理所当然的了。但是,输出误差 $\varepsilon(k)$通常是模型参数的非线性函数,因此在这种误差准则意义下,辨识问题将归结成复杂的非线性最优化问题。比如,如果图 1.7(a)中的模型取脉冲传递函数形式

$$\mu: G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} \tag{1.26}$$

式中

$$\begin{cases} A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a} \\ B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b} \end{cases} \tag{1.27}$$

则输出误差为

$$\varepsilon(k) = z(k) - \frac{B(z^{-1})}{A(z^{-1})} u(k) \tag{1.28}$$

且误差准则为

$$J(\theta) = \sum_{k=1}^{L} \left[z(k) - \frac{B(z^{-1})}{A(z^{-1})} u(k) \right]^2 \tag{1.29}$$

显然,误差准则函数 $J(\theta)$关于模型参数空间是非线性的。由于在确定这种情况的最优解时,需要用梯度法、牛顿法或共轭梯度法等迭代的最优化算法,这就使得辨识算法变得比较复杂。在实际应用中是否采用这种误差准则要视具体情况而定。

1.4　辨识的内容和步骤

辨识问题分为模型结构辨识和参数辨识(或估计)。当系统模型结构根据工程经验或采用模型结构的辨识确定后,主要的问题是模型的参数估计。从以上分析可见,系统辨识主要包括以下内容和步骤:

1. 设法取得系统输入输出的观测数据

为了获得一个最大可能接近实际对象特性的数学模型,在条件许可的情况下,要进行辨识实验。我们希望由辨识实验所获得的系统输入输出数据尽可能多地包括对象的信息。辨识实验设计包括:

1) 设计准则

为了对不同的试验方案进行比较,必须给出一个度量试验优劣性的准则。比如参数估计的精度,参数估计量一般是一个随机变量或随机向量,其估计的精度可以用一个估计误差的方差或协方差阵来衡量。数理统计中有下面的重要结论:对于一次完成的采样,参数估计的 $\hat{\boldsymbol{\theta}}$的协方差满足 Cramer-Rao 不等式

$$\mathrm{cov}\{\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}\} \geqslant \boldsymbol{M}_{\boldsymbol{\theta}}^{-1} \tag{1.30}$$

式中,$\boldsymbol{M}_{\boldsymbol{\theta}}$ 为 Fisher 信息函数矩阵,

$$\boldsymbol{M}_{\boldsymbol{\theta}} \triangleq \boldsymbol{E}_{\boldsymbol{\theta}} \left\{ \left[\frac{\partial \ln p(z \mid \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right]^{\mathrm{T}} \left[\frac{\partial \ln p(z \mid \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \right\} \tag{1.31}$$

式中，z 表示测量数据列；$\ln p(z|\boldsymbol{\theta})$表示给定参数 $\boldsymbol{\theta}$ 时希望 z 的条件概率密度的自然对数。根据式(1.30)，希望 $\boldsymbol{M}_{\theta}^{-1}$ 尽可能的小，使得参数估计的精度有可能提高；又根据式(1.31)，$\boldsymbol{M}_{\theta}$ 又依赖于测量数据，所以 $\boldsymbol{M}_{\theta}^{-1}$ 依赖于测量数据 z。因为 $\boldsymbol{M}_{\theta}$ 是一个方阵，度量的准则一般应为标量函数。在实验中，通常取

$$J = -\ln\det\boldsymbol{M}_{\theta} \tag{1.32}$$

作为设计准则。因为测量数据(包括所有的输入输出)与输入信号的选择有关，我们将推出各种关系的表达式，并据此设计出最优输入信号。

2) 持续激励输入信号的设计

用于辨识模型的输入信号通常要求是零均值的伪随机信号。无论采用何种信号，都要求信号是持续激励的。

3) 采样间隔的设计

在实验之前必须适当地选择采样间隔或输入信号的时序脉冲宽度，其主要依据是实际系统的采样间隔和所要求的模型精度。采样间隔 Δ 太大，会影响辨识精度。否则会增加存储量和计算量。一个经验的规则是

$$T_{gc}/\Delta \approx 5 \sim 15 \tag{1.33}$$

式中，T_{gc}是系统过渡过程时间的 95%；另外一规则是

$$\Delta \approx T_{\min} \tag{1.34}$$

或

$$\Delta \approx (0.05 \sim 0.1)T_a \tag{1.35}$$

式中，$T_{\min}$是对象的最小时间常数，T_a 为系统的主要时间常数。

2. 应有一个合适的模型集

为了所辨识的系统能从这个模型集中选择出一合适的模型，模型集可根据机理所得到的一些未知参数的模型结构，或是待定参数仅作为数据拟合工具的黑箱模型结构，也可以是根据系统实际工艺要求或系统预测所得到的一组待拟合曲线数据结构。

3. 必须有一个对辨识所得到模型的验证评价

对辨识所得到模型的验证是系统辨识的重要环节。验证的目的是为了确定该模型是否是模型集中针对当前观测数据的最佳选择。验证的方法主要有

(1) 利用先验知识验证，即根据对系统已有的知识来判断模型是否实用。

(2) 利用数据检验。当利用一组数据辨识得到一个模型之后，通常希望用另一组未参与辨识的数据检验模型的适用性。如果检验结果失败，可能存在的问题是：辨识所用的一组数据包含的信息不足或所选模型类不合适。另外，也可以用同

一组数据对不同模型进行比较,以选用更合适的模型。

(3) 利用实际响应检验。比较实际系统和模型的阶跃响应或脉冲响应是判别模型是否适用的重要手段。

(4) 利用激励信号$\{u(k)\}$的自相关函数校验。若验证$\{u(k)\}$是零均值的白噪声序列,相应的模型是可靠的,其置信度为95%。

1.5 典型的非线性系统辨识与控制方法

1.5.1 非线性辨识典型模型及辨识、控制方法特点

建立描述非线性现象模型是研究非线性问题的基础,Billings 对以前的非线性系统辨识结果做了一次总结性的综述。Titterington 和 Akitsos 在 1989 年也做了非线性系统辨识的综述。描述非线性系统的典型模型及对非线性辨识、控制的方法有:

1) Volterra 级数

Volterra 级数可表示为

$$y(t) = \sum_{n=1}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}h_n(\tau_1,\tau_2,\cdots,\tau_n)\prod_{i=1}^{n}u(t-\tau_i)\mathrm{d}\tau_i = \sum_{n=1}^{\infty}y_n(t) \tag{1.36}$$

函数 $h_i(\tau_1,\tau_2,\cdots,\tau_i)$ 是 Volterra 级数的核(kernel)。Volterra 的核(特别是频域核)有明确的物理意义。从模型辨识的角度看 Volterra 级数有一个明显的缺点,即需要相当多的被估计参数才能取得满意的精度。例如,当用 Volterra 级数去逼近一个二阶的非线性系统需要10^{10}个参数。Billings 对这个模型做过详细的研究,提出了几种辨识算法,尽管 Volterra 级数对非线性系统理论、函数逼近、辨识方法的发展有非常重要的推动作用,但认为它很难用过程建模。

2) NARMAX 模型子集

谢菲尔德大学的专家对非线性的辨识做了一次综述,他认为,NARMAX 模型

$$\begin{aligned}y(k) = F[&y(k-1),\cdots,y(k-n_a),u(k-1),\cdots,u(k-n_b),\\&e(k-1),\cdots,e(k-n_e)+e(k)]\end{aligned}$$

式中 $F(\cdot)$是一个非线性函数,$e(k)$是一个不可观测的零均值和有限方差的独立噪声,$k=0,1,2,\cdots$是离散时间标量。近年来,Sontag、Billings 和 Chen、Billings 和 Zhu 又找到了另一 NARMAX 模型子集,称为有理(rational)NARMAX 模型,其表达式为

$$y(k)=\frac{a(k)}{b(k)}+e(k)=\frac{\sum_{j=1}^{\text{num}} p_{nj}(k)q_{nj}}{\sum_{j=1}^{\text{den}} p_{dj}(k)q_{dj}}+e(k) \tag{1.37}$$

NARMAX模型提供了一个统一的有限可实现非线性系统表达式,像双线性型、Wiener模型、Armax模型等,其优点是逼近精度高,收敛速度快,对线性参数的子集模型辨识简便。可以用线性最小二乘法进行选项和估计参数。在国外已用在化工领域、海洋工程、电力工程。但当被测对象是多变量系统且阶数较高时,模型中的参数非常多。而且在这种模型中,模型结构辨识问题一直未解决。

3) 神经网络(neural network)模型

神经网络近年来得以飞速发展,并已有效地用于非线性系统的辨识和控制。目前广泛使用的有两种神经网络,一种是多层(multi-layer)神经网络,另一种是循环(recurrent)神经网络,从辨识的角度看,多层神经网络代表了静态非线性模型,而循环神经网络代表了动态非线性模型。Narendra等人提出了多层前馈网络用于非线性辨识的一般框架,Chen和Billings等从基于人工神经网络的Narma模型出发,进行了类似的研究,提出了并行递推预报误差辨识方法。另外,模糊控制和神经网络结合产生的模糊神经网络(FNN)也可以用于非线性辨识。

比其他非线性辨识方法优越的是神经网络可以不依赖模型函数,也就是说,可以不用了解被辨识非线性系统(被测系统)输入和输出之间存在何种数学关系。目前用的较多的是具有反传的前馈BP网络。只要给定系统输入样本、网络的结构以及系统输出的教师信号,给网络一组输入样本,便可得到对应的网络输出,利用网络输出和教师信号的差值来修正网络的权值和阈值,直至满足要求。换句话说,利用多层网络所具有的对任意非线性映射的逼近能力,来模拟实际系统的输入输出关系。但BP算法存在着局部极值和收敛速度慢等无法克服的缺点,这就促使人们去研究其改进的算法。我们对一类非线性系统神经网络辨识作过几种尝试性的改进算法研究,收到了较好效果,其中包括降低一类神经网络灵敏度的理论和方法、提高一类神经网络容错性的理论和方法、提高神经网络收敛速度的一种赋初值算法、改进适应度函数的遗传神经解耦控制器、隶属函数型神经网络与模糊控制融合的方法等[52~55,58~66]。这部分内容在第7章展开讨论。

4) H_∞控制理论

应用H_∞控制理论设计的控制器对各种摄动(外部干扰信号,系统内部参数变化,传感器噪声等)的灵敏度最低。这种设计可以使SISO系统达到最优,但对MIMO系统设计的控制器阶数太高,难以实现。但日本的华人专家Sheng Tielong等学者应用H_∞控制理论对MIMO系统的设计和对非线性系统控制做了大量深入的研究,认为,H_∞控制理论对于非线性系统控制具有较好的前景。

5）扩展的卡尔曼滤波

扩展的卡尔曼滤波可以用于非线性系统的状态估计。尽管贝叶斯(Bayes)估值给出了一种直观的精确表达式，但其在每一步的积分运算都是非常困难的，因而很难用于实际计算非线性的状态估计。扩展的卡尔曼滤波实际是把非线性模型进行线性化处理，然后利用卡尔曼滤波进行状态估计。用一阶近似扩展的卡尔曼滤波的精度与系统模型的非线性特性以及噪声水平有关。当非线性函数参考点附近比较平直并且信噪比较高时，该算法一般可获得足够精确的结果。为考察系统精度对系统进行仿真时，如果出现算法发散或精度较差时，可以考虑用二阶近似扩展的卡尔曼滤波方法。

6）微分几何法

微分几何法是研究非线性系统的一种新的工具，其作用相当于 SISO 线性系统的拉氏变换和多变量系统中的线性代数。近年来，非线性系统的几何理论已初步形成，用微分几何方法可以有效的解决非线性系统的能控性、能观性、可逆性、解耦、最小实现及系统的对称性、相似性等问题。

另外，预测(PC)、模糊解耦(FD)、滑模变结构(SMVS)、模型参考自适应(MRAC)、专家控制(EC)或相互的组合的方法均可用于非线性系统的控制。

1.5.2　非线性系统参数估计的特点

参数估计是在模型阶或模型的结构已确定后，根据系统所外加的输入样本及实测到的数据求模型中的参数（或模型的数学表达式中与各项有关的系数）。从参数估计的角度看，非线性的模型可分为线性参数模型和非线性参数模型。

系统模型为离散化的方程，其中输出变量中存在大于或等于二次以上的项，其他参数、输入都是线性的，该模型是线性参数的非线性模型。如果模型的参数是非线性的，且输入输出都是非线性的，这种模型是非线性参数的非线性模型。对参数估计来说，后者比前者复杂得多。

目前参数估计用的较多的方法是有预测误差(prediction error)算法，Jackoby 和 Pandit，Billings 和 Chen 已将这个算法用于估计非线性模型的参数；各种改进的最小二乘：Huffel 和 Van de Walle 基于线性变量有误差模型推导了一类改进最小二乘算法，但现实仍未推广到非线性模型的参数估计；Korenberg，Billings，Liu，Mcllroy 将直交最小二乘算法推广到线性参数的非线性模型的参数估计；Billings 和 Zhu 又将这一算法进一步推广到有理 NARMAX 模型(非线性参数)的参数估计，这一结果是非线性参数估计及变量有误差问题研究的一次大的进展。研究非线性模型的参数估计是研究非线性问题的一个重要方面。

1.5.3 神经网络及其系统控制结构

上面讨论了神经网络在非线性辨识中特点，这部分主要分析非线性系统神经网络控制的研究问题。非线性系统控制是当前自动控制领域一个难点。而基于神经网络的非线性系统控制是非线性控制研究中的一个热点。这是由于现代控制理论是以完备的系统模型为基础，使得将其用于非线性系统的建模相当困难。早在人工神经网络研究的初期，就已开始了控制领域应用的研究，但迄今为止，人工神经网络辨识方法还不能算作一种独立的成熟的方法，只是在原有的非线性控制方式中用神经网络模型替换原有模型，且将网络的训练过程与模型参考自适应、辨识过程结合使用。应根据被辨识对象的特性探讨出实用的算法。作者在神经网络辨识方面尝试过几种较有效的改进型的算法[52~55,58~66]，在第 7 章展开讨论。

下面简要提供几种学者们研究的神经网络控制结构形式及其现状：

1) 预测控制

预测控制又称为预估控制。Erramilli W Willinger、Scokaert P O M 和 Clarke D H 等证明：采用时段后退技术(receding horizon)可构成一种计算量有限的反馈控制算法，且这种算法可用于非线性系统，使系统达到稳定。在这种算法中，一般由系统输出和预测的方差值、系统本时刻输入和上一时刻输入的方差值组成一个二次型指标函数。根据系统前一时段的输出、输入数据，可由网络来预测后一时段系统的输出，经过优化处理，即可得到使指标函数最小情况下的控制信号。实际上预测的目的是给系统控制器的设计提供参考依据。

这种方法已成功地用在很多领域，如对大量的通信网络业务量的处理，学者们是通过神经网络训练，掌握通信业务源的内在规律，从而实现实时、动态地对业务源预测，其结果有助于对异步转移模块(ATM)网络的控制；船舶横摇动的时间序列预报；气象预报、工业过程控制等方面。

2) 控制系统的状态监测与故障诊断

故障诊断实质上是模式分类的过程。利用监测空间到故障空间的映射关系以及对系统在各时刻采集到的状态变量，判断系统运行正常与否。Timo Soraa 等指出，用于控制系统监测与故障诊断的神经网络主要有 MLP(多层感知器)神经网络、RBF(径向基函数神经网络)、ART(自适应共振)神经网络和 SOM(自组织映射)神经网络等。后两种非监督学习神经网络对于故障诊断准确率不高，如 ART 的误诊断率超过 30%。相比之下，MLP 网络能够对训练样本可靠分类且测试结果令人满意。

文献[33]提出系统状态监测与故障诊断方法很有代表性。基本步骤是：从实际控制系统提取监测状态变量构成监测向量 P；将 P 作为原始输入样本，根据函数型连接思想对 P 进行样本模式特性增强，形成 X；根据 X 和相应的监督信号对

神经网络进行离线训练,从而获取神经网络的结构参数;在线对系统状态监测与故障诊断时,利用已训练好的神经网络对系统监测的状态变量进行泛化;显示系统的运行状况正常与否。

在实际控制中特别是被控对象具有非线性特性时,人们往往无法得到被控对象的解析模型,更无望依据古典或现代控制理论设计有效的控制器。因而,现场技术人员常根据现场工况和积累的经验对系统进行控制。在这种情况下,可采用监督控制方式。这种控制方法已被用于智能控制的决策系统,倒立摆平衡控制等。

3) 模型参考自适应控制

在模型参考自适应控制系统中,系统期望的输出特性由一个能满足系统要求的参考模型给出,控制器在被控对象之前且处于串联连接的关系。基于神经网络的非线性系统模型参考自适应控制是Narendra等提出来的,将神经网络用作控制器,利用被控对象的输出和网络的差值来训练网络,网络的输出又作用给被控对象,使对象的输出逐渐逼近参考模型的理想输出,最终使系统达到平衡稳定。

4) 内模控制

在内模控制中,被控系统和一个系统的前向模型处于并列位置,被控对象的输出与该模型的差值作为反馈信号送到被控对象前(控制器和被控对象串联连接)的控制器,该系统要求控制器函数是被控对象的逆。由于内模控制有较强的鲁棒性,因而在过程控制中得到了广泛的应用。目前已研究智能自适应内模控制系统,如文献[26]将自适应内模控制系统在线调试成功,他用的是控制器参数模糊调整内模控制系统结构。用神经网络已得到的系统模型实现非线性系统的内模控制方法是Hunt等提出的[25]。内模控制器参数在线调整是根据被控对象的输出与神经网络模型的差值进行,故该神经网络必要是离线或在线辨识所得到的模型,它一定要能反映实际系统的特性。

5) 模糊神经网络系统

模糊神经网络系统从结构上看主要有两类:一是在神经网络结构中引入模糊逻辑,使其具有直接处理模糊信息的能力,把一般神经网络中的加权求和转为模糊逻辑运算中的"∨","∧"("析取:取大、即并","合取:取小、即交"),从而形成模糊神经元网络;二是直接利用神经网络的学习功能及映射能力,去等效模糊系统中的各个模糊功能块,如模糊化、模糊推理、模糊判决。另外,还可以把神经网络和模糊控制用在同一个系统中,以发挥各自的特长。

J.R.Jang采用暂存反向传播算法实现模糊神经网络的自学习[27]。这种方法利用自适应神经网络实现了时间序列预报和系统辨识,并形成了模糊控制器。倒立摆实验结果证明了这一方法的有效性和控制器的鲁棒性。神经网络在模糊自适应控制器中的另一个应用是改进和增强自组织模糊控制器的学习能力。T.Yamaguchi等提出用模糊联想记忆系统(FAM)实现自组织模糊控制的方法[28]。FAM

利用模糊规则表达专家知识,且采用神经网络的自学习能力来提炼知识。这样可在模糊规则的前提部分产生期望的隶属函数分布,在结论部分可单独训练控制器,以保证在特定的限制条件下达到最佳。从两个可调参数的模型(直升机飞行控制器)仿真研究发现,学习后直升机稳定飞行的效果明显优于学习前的效果。大多数的模糊神经网络系统被看做分层前馈网络,然后用 BP 网络来实现。Junbong Nie 等[29]提出一种不同的方法,通过引入局部网络结构和前馈推理算法,用整个 RBF (radial basis function)网来表达基于规则的模糊知识,并将其发展成多变量自组织和自学习模糊控制器。同时引入模糊竞争机构和重复学习控制算盘,使得基于模糊的 RBF 网(FRBF 网)的模糊控制器在缺少"专家知识"(或"教师信号")的情况下,能自组织其结构、自学习控制规律,以达到满意的控制效果。

在实际应用中,基于神经网络的自适应模糊控制器存在的主要问题是:由于神经网络的结构复杂、计算量大,使得控制器的计算时间过长。如何用简单的方法对模糊控制器进行量化、并转换成易于学习的算法;借用何种学习算法,如何确定学习指标,构成有效的模糊控制学习系统;如何用神经网络表达某种模糊模型等,都是有待进一步研究的问题。

1.5.4 非线性解耦问题

耦合是生产过程动态特性普遍存在的一种现象。若被控对象存在耦合,势必降低控制系统的调节性能。耦合严重时,会使系统无法正常运行。因此对解耦问题的研究无论是对控制理论的发展还是对工程实际都是非常有意义的。

目前有许多方法用于多变量控制的解耦问题,在工程上应用最多的解耦方法是 Bokesenbom Hood 提出的对角矩阵法和 Bristal 提出的相对增益法,它们均是基于过程通道的传递函数来设计解耦控制器,使系统的零极点对消的方法,可得到简便形式的控制器,但这些方法使得被控对象的动态特性发生变化,而且变得相对复杂些,从而使控制器的设计有一定困难。前馈补偿解耦控制也有用到工程中的,前馈补偿解耦实际是把某通道的调节器输出对另外通道的影响看做是扰动作用,然后应用前馈控制的原理消除回路的耦合,以上解耦方法均要求系统模型确定。而非线性系统多属参数易变,模型不确定。因此上述解耦方法不应用于非线性系统的解耦。

尽管解耦理论已取得了不少成果,但与最优控制、自适应控制等其他分支相比,解耦理论在工程的应用确不能令人满意。

目前,利用模糊理论对多变量非线系统进行控制或解耦控制也是研究非线性问题的一个热点。国内外对多变量模糊控制系统的研究方兴未艾,逐渐成为模糊领域研究的热门课题。多变量模糊控制系统主要有:分层多变量模糊控制器、自学习模糊控制器、基于模型的多变量模糊控制方法、多变量的模糊解耦以及基于神经

网络的模糊控制。

模糊解耦的结构与特点大致包括：

1）直接模糊解耦

这种方法是对被控对象和解耦补偿器相串联，先进行解耦，然后针对解耦后的各变量进行模糊控制设计。该方法是徐承伟和吕勇哉首先提出来的，给出了实现解耦的一个充要条件，但解耦补偿器的结构和参数用经验试凑法离线确定的，没有通用法，很难实现完全解耦[36]。随后，他们又提出模糊关系系统的反馈解耦考虑到用于多输入多输出（MIMO）系统的情况，文中给出了实现解耦的一个充分条件，但控制器的构成尚缺少理论依据[37]。

2）间接模糊解耦法

Chen 等引入相关因子，研究出一类多变量模糊控制器。Czogalaand 和 Zimmermann 引入随机相关因子，利用此类因子构造出多维概率模糊控制器。Gupta 等在此研究基础上，提出通过对多变量模糊控制规则进行子空间的分析，然后用一组二维模糊方程描述多维模糊控制规则。文献[40]对 Gupta 解耦算法中采用的 Mamdani 推理合成规则进行了改进，提出一种新算法，将“max-min”合成算子修改为“max-∧”运算从理论上证明了采用新的合成算子所得到的推理结论更为确定，且满足一致性条件。

总之，模糊解耦控制系统的研究尚处于发展阶段，很多结论只是理论推导，还不能进入实验室进行验证。同时，针对解耦之后控制系统的稳定性、可控可观性的研究也尚未成型。因此，无论是针对控制对象还是针对控制器解耦，当前急需要解决的问题是如何由模糊关系方程求解各个解耦后的模糊子关系，尽量减少前述方法所加的约束条件，得到令人满意的仿真效果。

1.5.5　需要深入研究的非线性问题

智能控制是传统控制理论发展的高级阶段，主要用来解决那些用传统方法无法解决的复杂系统（非线性系统）控制问题。前述的神经网络及模糊控制均属智能控制范畴，即是智能控制的一个分支。智能控制中也有不少需要辨识模型的问题。全球第一届智能控制会议从 1993 年 8 月在北京召开以来，相继于 1997 年在西安交通大学、2000 年在合肥、2002 年在上海和 2004 年在杭州召开了四届全球智能控制会议，会上全球关注智能控制的专家研讨智能控制的热点问题。今后如何按人们所关心的问题去研究，从智能控制应用发展看，应从以下几方面研究：

1）网络理论方面

在发展人工神经网络理论中，要突出其学习、并行处理、联想记忆功能。进一步开发 Petri 网及其他网络功能，因为这些网络在系统建模时已脱离了传统的方法，给智能控制带来了生机。尤其是在神经网络逼近非线性函数（特性）及其控制

系统方面应解决如下问题。

(1) 对不同的逼近对象，如在辨识中对不同的被辨识非线性模型，如何选择神经网络结构，当选择前馈多层网络时，问题就落实到如何确定网络的总体结构形式、网络层数、以及每层节点的个数，目前还缺乏理论指导。尽管有不少学者投身到该问题的研究中，但只停留在经验上和启发式的规则上，问题有待深入研究。

(2) 从如何提高网络的训练速度上深入研究。要从网络作用函数的映射机理、网络的灵敏度、网络的容错能力以及抗干扰性等问题上深入研究，找出现有网络逼近复杂非线性特性速度慢的根本原因，进而研究提高训练速度的理论和方法。

(3) 从提高神经网络系统的控制精度方面研究。神经网络作为控制器对系统进行实时控制时，由于网络的训练速度限制，加之非线性系统的复杂性，使得这一问题具有相当大的难度。对于这种系统的稳定性、可控性理论还需深入研究。另外，由于非线性系统的噪声是普遍存在的，因此研究如何提高网络抗干扰性、降低灵敏度及提高神经网络控制器的鲁棒性问题是提高神经网络系统的控制精度的关键，须进一步深入研究。

另外，神经网络发展的最终要以大规模的高精度的并行计算集成电路(硬件)来实现，因此，神经网络的硬件开发及其研究也是一个研究神经网络的重大发展方向。若这个问题能解决，将对神经网络控制复杂的非线性系统的控制速度及精度提高是一个很大的促进。

2) 模糊控制理论

(1) 模糊集理论的研究。模糊集理论是介于逻辑计算与数值计算之间的一种数学工具，其形式上利用规则模糊推理。这正补偿了人工神经网络的弱点。

(2) 从多变量模糊控制的理论方面研究。其中有模糊规则设计方法，包括隶属函数的确定、量化因子、采样周期、规则系数的最优选择、规则和隶属函数的自动生成等；多维模糊推理方法的研究，以满足高性能多变量控制器设计的需求；多变量控制系统稳定性的研究，以及对可控性、可观性等性能指标的评估方法；模糊动态模型的辨识方法及相应的自学习方法；基于模型的模糊自校正控制系统的研究；神经网络与模糊控制相结合，从而提高系统控制精度方法的研究；实用多变量控制技术的开发与应用，为解决像多自由度的机器人柔性运动控制这样的复杂问题服务。重要发展动向是 MIMO 模糊控制器的集成化和 MIMO 模糊计算机的开发。

3) 信息融合理论及方法的研究

一个复杂的多变量非线性系统要得到较理想的控制效果，多个传感器多种信息的融合是关键。除了传统的基于 Bayes 参数估计数据融合方法外，应对基于模糊理论信息融合方法、粗糙集理论与智能方法结合的信息融合方法进行研究。

4) 数据挖掘

数据挖掘方法是对复杂的多变量非线性系统研究的重要方法之一。支持向量

机技术可经济、有效的挖掘所需的甚至是无法测到的(野点)数据。

5) 知识工程的研究

尤其作为智能控制重要分支的专家系统、学习系统都离不开知识的运用、知识的获取和更新,因而知识工程应进一步研究。

6) 系统可靠性能理论的研究

一个智能控制系统的正常运行,离不开故障诊断,故障的随时检测与自动排除,以及一些算法中的冗余、容错理论的研究对系统控制的可靠性至关重要。

7) 混沌的诱导与控制

非线性控制系统的混沌研究也是对控制学科学者们的新挑战。20世纪70年代,美国大气物理学家 Lorenz 在实验室里观察到一个完全确定的非线性方程组(现被称为的 Lorenz 方程),对初值异常敏感,具有貌似随机信号的输出。十多年后,许多学者相继在不同的领域中发现了这种现象。Li 和 Yorke 首先称之为“混沌(chaos)”。这在科学界引起了一场很大的振动,有的学者认为,混沌现象的发现否定了传统的确定论时空观,将是20世纪量子力学和相对论之后第三次技术观念的革新。广泛地说,器件的非线性是绝对的,而线性是相对的。线性状态只是非线性状态的一种近似或一种特例而已。由于混沌运动是非线性系统的一种普遍的运动,模糊了传统的确定系统和随机系统的界限,目前是最活跃的科学领域之一,已成为数学、物理、化学、力学、光学、天文学及脑科学等重多学科的重点研究课题。对这种新的、复杂的混沌对象的研究,传统基于小扰动。线性化的处理方法不再适用,具体的学科的一些定义、定理可能将改变(如控制理论中系统的稳定分为:常值稳定、周期稳定、混沌状态)。混饨的特征:对初值异常敏感,不能进行长期预测,确定性系统却有貌似随机的响应,局部不稳定,但是总体吸引,与常规运动(平衡周期运动、准周期运动)不同,是一种始终限于有限区域并且轨迹不重复性态复杂的运动。文献[71]指出:“控制系统的混沌运动来自三个方面:一是非线性控制系统本身的混沌;二是控制算法带来的混沌;三是系统离散化引起的混沌。若系统模型用状态方程来描述,对定常系统而言,只有三阶以上的系统能够产生混沌。”近年来有关专家对混沌识别与混沌系统辨识、混沌系统的诱导与控制进行了深入的研究,这对非线性问题的稳定控制开辟了新的途径。有关反馈动态系统出现的混沌现象、基于控制理论的混沌分析方法、混沌识别与混沌系统辨识问题在第8章展开。

1.6 小 结

本章是系统和模型及辨识建模的基本概念,其中1.1、1.2节包括模型的表现形式及数学模型的分类、辨识的定义;1.3节介绍了辨识的问题的表示形式及原理,其中包括辨识问题的表达形式、辨识算法的基本原理和误差准则;1.4节是辨

识的内容和步骤;最后,1.5 节介绍了典型的非线性系统辨识与控制方法,其中包括非线性系统辨识与控制方法的特点、非线性参数估计的特点、神经网络及其控制系统结构、非线性解耦及需要深入研究的非线性问题。

习　题

1. 简述数学模型的表现形式及其分类。
2. 辨识建模的定义是什么?
3. 辨识问题的表达形式是什么?
4. 什么是辨识的误差准则?
5. 简述辨识的内容和步骤。
6. 神经网络非线性辨识的主要特点是什么?

第 2 章　辨识理论基础及古典辨识方法

本章是系统辨识基础，包括辨识的理论基础和古典辨识法两部分内容。2.1～2.4 节主要介绍随机过程的基本概念及其数学描述、线性系统在随机输入下的响应等，并讨论随机序列和白噪声及其产生方法。系统辨识所用的数据通常含有噪声，它的基本概念及其产生方法与系统辨识密切相关。因此，分别讨论并剖析用 MATLAB 语言开发的产生随机序列、白噪声和 M 序列的三种程序。在 2.5 节古典辨识法中，介绍相关分析法辨识和相关函数的概念，讨论相关分析频率响应和相关分析脉冲响应法辨识，并列举相关分析脉冲响应法辨识的实例。

2.1　随机过程基本概念及其数学描述

2.1.1　基本概念

在科学技术领域中，人们观察到各式各样的事物变化过程。有些变化过程具有明确的规律性，如自由落体运动、电容充电过程等，这些称为确定性过程。还有些变化过程具有偶然性，例如电子放大器的零点漂移、风浪中海面的起伏等，人们无法预知下一时刻将会发生什么情况，这些称为随机过程。对于随机过程，我们可以在完全相同的条件下(在人们力所能及的意义上)进行多次测试，这样就得到很多样本，它们的变化过程互不相同。样本具有偶然性，但他们的总体却往往具有统计意义上的规律性。按照严格的定义，所谓“随机过程”就是指大量样本 $x_1(t)$，$x_2(t)$，…所构成的总体。

1. 随机过程数学描述

为了对随机过程 $x(t)$进行数学描述，应该注意以下几点：

(1) 在每一孤立的瞬间，$x(t)$的取值是随机的，它是一个随机变量。因此，$x(t)$首先可以用描述一维随机变量的方法加以描述，这就引出随机过程一维概率密度 $p_1(x,t)$的概念。

(2) 沿时间坐标轴看，$x(t)$是 x 的取值随时间变化的过程。为了完整的描述一个随机过程，还需反映不同的时刻 x 取值之间的联系。二维概率密度 $p_2(x_1,x_2;t_1,t_2)$代表 x 在 t_1 时刻取值为 x_1、t_2 时刻取值为 x_2 的概率密度。这就是说，$p_2(x_1,x_2;t_1,t_2)\mathrm{d}x_1\mathrm{d}x_2$ 代表 $x(t)$相继通过随机过程图 2.1 中 $\mathrm{d}x_1$ 和 $\mathrm{d}x_2$ 两个小窗口的概率。

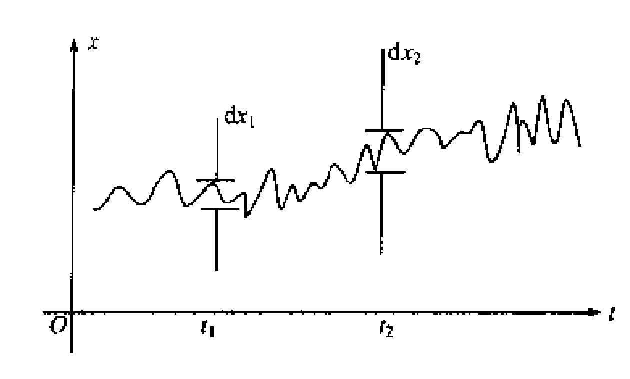


图 2.1　$p_2(x_1,x_2;t_1,t_2)dx_1dx_2$ 概率

(3) 严格意义上说,要真正完整地描述 $x(t)$,除一维、二维概率密度外,还需要三维、四维……概率密度。这几乎是无法满足的苛刻要求。在实际应用中,人们不得不满足于用某些数字特征来近似地刻画一个随机过程。

2. 随机过程的特征

在实际应用中,人们只限于用一些最基本的数字特征来刻画 $x(t)$,即 $p_1(x,t)$和 $p_2(x_1,x_2;t_1,t_2)$有关的两类数字特征。

如果平稳随机过程 $x(t)$的各集合平均值等于相对应的时间平均值,即

$$\begin{cases}\bar{x} = \mu_x \\ \overline{x(t)x(t+\tau)} = R_x(\tau)\end{cases} \tag{2.1}$$

式中,$\bar{x}$ 为随机过程$x(t)$的时间均值(指对 dt 的积分的均值);μ_x 为与一维概率密度 $p_1(x,t)$有关的数字特征量集合均值(指对 dx 的积分的均值);$R_x(\tau)$为自相关函数。则称 $x(t)$是各态遍历的平稳的随机过程。

对于各态遍历的平稳随机过程,它的两个基本数字特征就可以只根据一个很长的样本按下式计算

$$\begin{cases}\mu_x = \lim\limits_{T\to\infty}\dfrac{1}{2T}\displaystyle\int_{-T}^{T}x(t)dt \\ R_x(\tau) = \lim\limits_{T\to\infty}\dfrac{1}{2T}\displaystyle\int_{-\infty}^{\infty}x(t)x(t+\tau)dt\end{cases} \tag{2.2}$$

实际上,只能使 T 取尽可能大的有限值,见图 2.2。假定 $T=NT_0$,$\tau=lT_0$ 为采样时间,则式(2.2)变为

$$\begin{cases}\mu_x \approx \dfrac{1}{N}\displaystyle\sum_{k=1}^{N}x(k) \\ R_x(\tau) \triangleq R_x(l) \approx \dfrac{1}{N-l}\displaystyle\sum_{k=1}^{N-l}x(k)x(k+l)\end{cases} \tag{2.3}$$

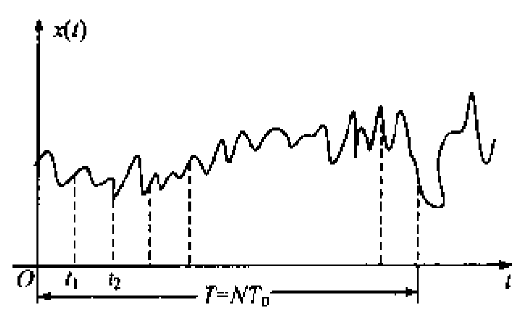


图 2.2　μ_x 和 $R_x(\tau)$的近似计算

以上只讨论了一个随机过程单独存在时的情况。有时在同一个问题中涉及两个互相有关的随机过程 $x(t)$和 $y(t)$,例如一个出现在过程的输入侧,另一个出现在输出侧,则可用互相关函数和互协方差函数来描述它们之

间的联系。

互相关函数

$$R_{xy}(\tau) \triangleq E\{x(t)y(t+\tau)\} \tag{2.4}$$

互协方差函数

$$C_{xy}(\tau) \triangleq \operatorname{cov}\{x(t), y(t+\tau)\} \triangleq \{[x(t)-\mu_x][y(t+\tau)-\mu_y]\}$$
$$= R_{xy}(\tau) - \mu_x\mu_y \tag{2.5}$$

若 $C_{xy}(\tau)=0, \forall -\infty<\tau<\infty$,则称 $x(t)$与 $y(t)$互不相关。

2.1.2 相关函数和协方差函数的性质

1) 自相关函数和自协方差函数具有以下性质

(1)

$$R_x(0) = E\{x(t)z(t+0)\} = \psi_x^2 \geqslant 0 \tag{2.6}$$

(2) 根据 $R_x(\tau)$的定义,显然有 $R_x(-\tau)=R_x(\tau)$,这说明 $R_x(\tau)$对称于纵轴。

(3)

$$R_x(\tau)| \leqslant R_x(0) \tag{2.7}$$

(4) 有时将设计的伪随机信号,如 M 序列,具有周期性,称为“周期性”随机过程。它们的自相关函数也具有周期性,即若 $x(t+T)=x(t)$,则 $R_x(\tau+T)=R_x(\tau)$。

(5) 若 $x(t)$均值不为零,则可分解成 $x(t)=y(t)+\mu_x$,式中,$y(t)$是均值为零的随机过程。此时

$$R_x(\tau) = E\{[y(t)+\mu_x][y(t+\tau)+\mu_x]\} = R_y(\tau) + \mu_x^2 \tag{2.8}$$

换言之 $x(t)$中的直流成分使其自相关函数向上平移 μ_x^2。

(6) 若出现 $x(t)$均值为零,且不含有周期性成分,则当 τ 很大时,$x(t+\tau)$与 $x(t)$必然是相互独立的。因而此时 $R_x(\tau)=0$。

(7) $x(t)=x_1(t)+x_2(t)$且 $x_1(t)$与 $x_2(t)$互不相关

$$R_x(\tau) = R_{x_1}(\tau) + R_{x_2}(\tau) \tag{2.9}$$

根据以上诸性质,可知 $R_x(\tau)$所具有的一般形态:它对称于纵轴,且在 $\tau=0$ 处取极大值(但并不意味着在该点有$\frac{\mathrm{d}}{\mathrm{d}\tau}R_x(\tau)=0$)。此外还可以从 τ 很大时曲线的形态判明 $x(t)$之均值是否为零,以及它是否含有周期性成分。

(8) 自协方差函数 $C_x(\tau)$与自相关函数 $R_x(\tau)$的基本关系是 $C_x(\tau)=R_x(\tau)-\mu_x^2$,因而 $C_x(\tau)$等于 $R_x(\tau)$向下平移 μ_x^2。显然,除此之外,它保留$R_x(\tau)$的形态不变。

2）互相关函数与互协方差函数的性质

（1）$R_{xy}(\tau)$既不是τ的偶函数，也不是τ的奇函数。换言之，互相关函数不具有对称性。

（2）$R_{yx}(\tau) \triangleq E\{y(t)x(t+\tau)\} \neq R_{xy}(\tau)$，但 $R_{yx}(-\tau) = E\{y(t)x(t-\tau)\} = R_{xy}(\tau)$。

（3）若 $x(\tau)$，$y(\tau)$中至少有一个均值为零，则 $C_{xy}(\tau) = R_{xy}(\tau)$。

2.2 谱密度与相关函数

2.2.1 帕塞瓦尔（Parseval）定理与功率密度谱表示式

帕塞瓦尔定理，常将帕塞瓦尔定理视为确定性过程的总能量谱表示式。若将确定性过程 $x(\tau)$视为流经 1Ω 电阻的电流，则 $\int_{-\infty}^{\infty} x^2(t)\mathrm{d}t$ 代表总能量。如果 $x(\tau)$的傅里叶变换$X(\mathrm{j}\omega)$存在，则可以证明总能量为

$$\int_{-\infty}^{\infty} x^2(t)\mathrm{d}t = \frac{1}{2\pi}\int_{-\infty}^{\infty} \| X(\mathrm{j}\omega)\|^2 \mathrm{d}\omega \tag{2.10}$$

上式称为帕塞瓦尔定理。它把总能量以频谱的形式表达出来。

证明 由于 $x(t)$和 $X(\mathrm{j}\omega)$构成一组傅里叶变换对，故

$$\begin{cases} X(\mathrm{j}\omega) = \int_{-\infty}^{\infty} x(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t \\ x(t) = \int_{-\infty}^{\infty} X(\mathrm{j}\omega)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}\omega \end{cases} \tag{2.11}$$

因此

$$\begin{aligned} \int_{-\infty}^{\infty} x^2(t)\mathrm{d}t &= \int_{-\infty}^{\infty}\left[x(t)\frac{1}{2\pi}\int_{-\infty}^{\infty} X(\mathrm{j}\omega)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}\omega\right]\mathrm{d}t \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\mathrm{j}\omega)\left[\int_{-\infty}^{\infty} x(t)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}t\right]\mathrm{d}\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\mathrm{j}\omega)X(-\mathrm{j}\omega)\mathrm{d}\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \| X(\mathrm{j}\omega)\|^2\mathrm{d}\omega \end{aligned} \tag{2.12}$$

2.2.2 维纳-辛钦（Wiener-Khintchine）关系式

可以证明，随机过程 $x(t)$的谱密度 $S_x(\omega)$与自相关函数 $R_x(\tau)$之间存在极其简单的关系——它们构成一组傅里叶变换对，即

$$\begin{cases} S_x(\omega) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j\omega\tau} d\tau \\ R_x(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_x(\omega) e^{j\omega\tau} d\omega \end{cases} \tag{2.13}$$

式(2.13)称为维纳-辛钦关系式。

考虑到 $S_x(\omega)$ 和 $R_x(\tau)$ 均为实偶函数,故式(2.13)可简化为

$$\begin{cases} S_x(\omega) = 2\int_{0}^{\infty} R_x(\tau) \cos\omega\tau d\tau \\ R_x(\tau) = \frac{1}{\pi} \int_{0}^{\infty} S_x(\omega) \cos\omega\tau d\omega \end{cases} \tag{2.14}$$

它是维纳-辛钦关系式的余弦变化形式。

如上所述,对于一个单独存在的随机过程 $x(t)$,维纳-辛钦关系式指明 $S_x(\omega)$ 与 $R_x(\tau)$ 构成一组傅里叶变化对。

如果在同一问题中出现两个随机过程 $x(t)$ 和 $y(t)$,则除它们各自的自相关函数和谱密度以外,还用互相关函数 $R_{xy}(\tau)$ 来描述两个随机过程之间的联系。此时可以设想把维纳-辛钦关系式加以推广,从而引出“互谱密度”的概念,定义互谱密度为互相关函数的傅里叶变换。由于 $R_{xy}(\tau)$ 并非偶函数,故它的傅里叶变换不是实函数而是复函数,将互谱密度记为 $S_{xy}(j\omega)$。因而存在以下关系

$$\begin{cases} S_{xy}(j\omega) = \int_{-\infty}^{\infty} R_{xy}(\tau) e^{-j\omega\tau} d\tau \\ R_{xy}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{xy}(j\omega) e^{j\omega\tau} d\omega \end{cases} \tag{2.15}$$

2.3 线性系统在随机输入下的响应

线性系统在平稳随机输入下,经过一段过度过程后,其输出 $y(t)$ 也是一个稳定的随机信号。线性系统的特性可用单位脉冲响应 $g(t)$ 和频率响应时间 $G(j\omega)$ 来描述。$g(t)$ 和 $G(j\omega)$ 构成一组傅里叶变换对,考虑到 $g(t)=0, \forall t<0$,则有

$$\begin{cases} G(j\omega) = \int_{0}^{\infty} g(t) e^{-j\omega\tau} d\tau \\ g(t) = \frac{1}{2\pi} \int_{0}^{\infty} G(j\omega) e^{j\omega\tau} d\omega \end{cases} \tag{2.16}$$

线性系统在随机输入下输出 $y(t)$ 的谱密度 $S_y(\omega)$ 与输入 $x(t)$ 的谱密度 $S_x(\omega)$ 之间有以下关系

$$S_y(\omega) = \| G(j\omega) \|^2 S_x(\omega) \tag{2.17}$$

系统在随机输入下,系统输入输出互相关谱密度 $S_{xy}(j\omega)$ 和输入谱密度

$S_x(\omega)$之间有以下关系

$$S_{xy}(j\omega) = G(j\omega)S_x(\omega) \tag{2.18}$$

比较式(2.17)与式(2.18),前者表明,对于系统的随机输入 $x(t)$,输出谱密度 $S_y(\omega)$只反映线性系统的幅值特性;后者表明,输入输出互相关谱密度 $S_{xy}(j\omega)$反映线性系统的动态特性。在随机输入情况下,通过谱密度分析可以确定未知线性系统的频率响应。

2.4 白噪声产生方法及其仿真

辨识所用的数据通常含有噪声。如果这种噪声相关性较弱或者强度很小,则可以近似将其视为白噪声。因此,白噪声是一类非常重要的随机过程,它的基本概念及其产生方法与系统辨识密切相关。

2.4.1 白噪声的概念

1) 白噪声的特点

白噪声过程是一种最简单的随机过程。严格地说,它是一种均值为零、谱密度为非零常数的平稳随机过程,或者说它是由一系列不相关的随机变量组成的一种理想化随机过程。白噪声过程没有"记忆性",也就是说 t 时刻的数值与 t 时刻以前的过去值无关,也不影响 t 时刻以后的将来值。如图 2.3 所示。

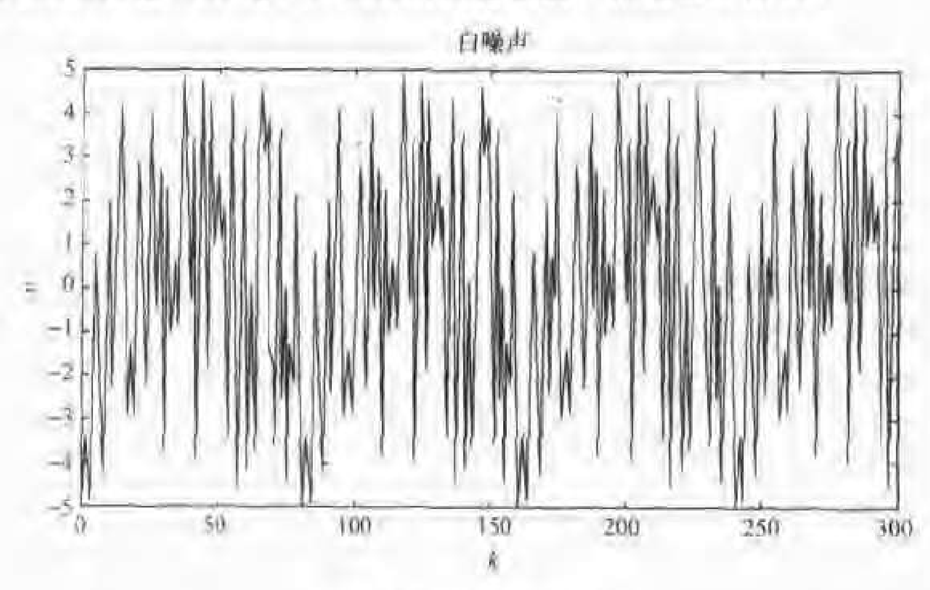


图 2.3 白噪声过程

白噪声过程可以在数学上描述如下:如果随机过程 $w(t)$的自相关函数为

$$R_w(\tau) = \sigma^2\delta(\tau) \tag{2.19}$$

式中,$\delta(\tau)$为 Dirac 函数,即

$$\delta(\tau) = \begin{cases} \infty, & \tau = 0 \\ 0, & \tau \neq 0 \end{cases} \tag{2.20}$$

且

$$\int_{-\infty}^{\infty} \delta(\tau)\mathrm{d}\tau = 1 \tag{2.21}$$

则称该随机过程为白噪声过程。注意在上述定义中已经包含了均值为零的概念。

由于 $\delta(\tau)$的傅里叶变换为 1,根据维纳-辛钦关系式可知白噪声过程 $w(t)$的谱密度为常数 σ^2,即

$$S_w(\omega) = \sigma^2, \qquad \infty < \omega < \infty \tag{2.22}$$

另外,服从于正态分布的白噪声过程称为正态(高斯)白噪声。

以上有关标量白噪声的概念可以推广到向量中。如果一个 N 维向量随机过程 $\boldsymbol{w}(t)$满足

$$\begin{cases} E\{\boldsymbol{w}(t)\} = 0 \\ \mathrm{cov}\{\boldsymbol{w}(t), \boldsymbol{w}(t+\tau)\} = E\{\boldsymbol{w}(t)\boldsymbol{w}^{\mathrm{T}}(t+\tau)\} = \boldsymbol{Q}\delta(\tau) \end{cases} \tag{2.23}$$

式中,$\boldsymbol{Q}$ 是正定的常数矩阵;$\delta(\tau)$是 Dirac 函数;则 $w(t)$为向量白噪声过程。

2) 白噪声序列

白噪声序列是白噪声的一种离散形式。根据傅里叶变换,白噪声的谱密度为常数 σ^2,即

$$S_w(\omega) = \sum_{l=-\infty}^{\infty} R_w(l)\mathrm{e}^{-\mathrm{j}\omega l} = \sigma^2 \tag{2.24}$$

式中,$R_w(l)$为两两不相关的随机序列$\{w(k)\}$的自相关函数

$$R_w(l) = \sigma^2\delta_l, \quad l = 0, \pm 1, \pm 2, \cdots \tag{2.25}$$

同样,向量白噪声序列$\{w(k)\}$满足

$$\begin{cases} E\{\boldsymbol{w}(k)\} = 0 \\ \mathrm{cov}\{\boldsymbol{w}(k), \boldsymbol{w}(k+l)\} = E\{\boldsymbol{w}(k)\boldsymbol{w}^{\mathrm{T}}(k+l)\} = \boldsymbol{R}\delta_l \end{cases} \tag{2.26}$$

式中,$\boldsymbol{R}$ 是正定的常数矩阵;δ_l 是 Kronecker 符号,即

$$\delta_l = \begin{cases} 1, & l = 0 \\ 0, & l \neq 0 \end{cases} \tag{2.27}$$

3) 表示定理

工程实际中数据所含的噪声往往是有色噪声。所谓有色噪声(或相关噪声)指的是噪声序列中每一时刻的噪声和另一时刻的噪声是相关的。对含有有色噪声的数据需要采用新的辨识方法才能得到满意的辨识结果。在特定情况下,有色噪声可以用白噪声来描述,这就是表示定理所要阐述的问题。

表示定理　设平稳噪声序列$\{e(k)\}$的谱密度 $S_e(\omega)$是 ω 的实函数或是 $\cos\omega$ 的有理函数,那么必须存在一个渐近稳定的线性环节,如果该环节的输入是白噪声

序列，则环节的输出是密度为 $S_e(\omega)$ 的平稳噪声序列 $\{e(k)\}$。

定理表明，有色噪声序列可以由白噪声序列驱动的线性环节输出来近似。该线性环节叫作成形滤波器，如图 2.4 所示。

白噪声 $\{w(k)\}$ → 线性系统（成形滤波器）$H(z^{-1})$ → 有色噪声 $\{e(k)\}$

图 2.4 成形滤波器

图中，$\{w(k)\}$ 是均值为零的白噪声序列，$\{e(k)\}$ 是有色噪声。成形滤波器的脉冲传递函数可以写成

$$H(z^{-1}) = \frac{D(z^{-1})}{C(z^{-1})} \tag{2.28}$$

$$\begin{cases} C(z^{-1}) = 1 + c_1 z^{-1} + \cdots + c_{n_c} z^{-n_c} \\ D(z^{-1}) = 1 + d_1 z^{-1} + \cdots + d_{n_d} z^{-n_d} \end{cases} \tag{2.29}$$

且 $C(z)$ 和 $D(z)$ 的根都在 z 平面的单位圆内。

2.4.2 白噪声的产生及其 MATLAB 仿真

如果在计算机上比较经济的产生统计上理想的各种不同分布的白噪声序列，则对系统辨识仿真研究提供了极大方便。为了简单起见，常把各种不同分布的白噪声序列称为随机数。从理论上讲，只要有了一种具有连续分布的随机数，就可以通过函数变换的方法产生其他任意分布的随机数。显然，在具有连续分布的随机数中，(0,1)均匀分布的随机数是最简单、最基本的一种，有了(0,1)均匀分布的随机数，便可以产生其他任意分布的随机数和白噪声。所以，下面着重讨论(0,1)均匀分布随机数及(−1,1)白噪声的产生及其仿真方法。

1) (0,1)均匀分布随机数的产生方法

在计算机上产生(0,1)均匀分布随机数的方法主要有三类。一类是把已有的(0,1)均匀分布随机数放在数据库中，使用时访问数据库，这类方法虽然简单但占用存储空间大；另一类是物理方法，用硬件实现；第三类是利用数学方法产生(0,1)均匀分布随机数，该方法经济实用，主要包括乘同余法和混合同余法。下面讨论简单实用的乘同余法。

用如下递推同余式产生正整数序列 $\{x_i\}$

$$x_i = Ax_{i-1} (\mathrm{mod}\ M), \quad i = 1,2,3,\cdots \tag{2.30}$$

式中，mod 为取 M 的余数，如：M 为 2 的方幂，即 $M = 2^k$，k 为 $k>2$ 的整数；若 $k=3$，则(mod 8)或 $k=8$，则(mod 256)，且 A 应取适中的值，如 $3<A<10$；初值 x_0 取正奇数，如 $x_0 = 1$。

再令

$$\xi_i = \frac{x_i}{M}, \quad i = 1,2,3,\cdots \tag{2.31}$$

可以证明序列$\{\xi_i\}$是伪随机数序列；同时还可以证明伪随机序列$\{\xi_i\}$的循环周期达到最大值2^{k-2}。

将式(2.30)和(2.31)合并为

$$\xi_i = \{A\xi_{i-1}\} \tag{2.32}$$

式中，初值为$\xi_0 = \frac{x_0}{M}$。

2) (0,1)均匀分布随机数的产生仿真举例

例 2.1　利用乘同余法，选$R=2, A=6, k=8, M=2^k=256$，递推 100 次，采用 MATLAB 的仿真语言(m 软件)编程，产生(0,1)均匀分布随机数。

解　(附光盘上的该程序：FLch2sjxleg1.m，可直接在 MATLAB 6.I 下运行。)

① 编程如下：

```
A=6;  N=100;     x0=1;  M=255;  %初始化;
for k=1:N                %乘同余法递推 100 次开始;
   x2=A*x0;              %x2 和 x0 分别表示 xi和 xi-1;
   x1=mod(x2,M); %将 x2 存储器的数除以 M,取余数放 x1 中;
   v1=x1/256;            %将 x1 存储器的数除以 256 得到小于 1 的随机数放 v1 中;
   v(:,k)=v1;            % 将 v1 中的数(ξi)存放在矩阵存储器 v 的第 k 列中,
                         % v(:,k)表示行不变、列随递推循环次数变化;
   x0=x1;                %xi-1 = x1,x1 中存放第 i 时刻的余数;
   v0=v1;
end                      %递推 100 次结束;
v2=v                     %该语句末无';',实现矩阵存储器 v 中随机数放在 v2 中
                         %且可直接显示在 MATLAB 的 window 中;
k1=k;
 %grapher                 %绘图;
k=1:k1;
plot(k,v,k,v,'r');
xlabel('k'), ylabel('v');tktle('(0-1)均匀分布的随机序列')
```

② 程序运行结果如图 2.5 所示。

③ 产生的(0,1)均匀分布的随机序列。

在程序运行结束后，产生的(0,1)均匀分布的随机序列，直接从 MATLAB 的 window 界面中拷贝出来如下(v_2 中每行存 6 个随机数，$v_2(j,k), j=1,2,\cdots,13$; $k=1,2,\cdots,6$)：

```
v2 =
    0.0234    0.1406    0.8438    0.0820    0.4922    0.9609
    0.7852    0.7266    0.3750    0.2578    0.5508    0.3164
    0.9023    0.4336    0.6094    0.6680    0.0234    0.1406
    0.8438    0.0820    0.4922    0.9609    0.7852    0.7266
```

0.3750	0.2578	0.5508	0.3164	0.9023	0.4336
0.6094	0.6680	0.0234	0.1406	0.8438	0.0820
0.4922	0.9609	0.7852	0.7266	0.3750	0.2578
0.5508	0.3164	0.9023	0.4336	0.6094	0.6680
0.0234	0.1406	0.8438	0.0820	0.4922	0.9609
0.7852	0.7266	0.3750	0.2578	0.5508	0.3164
0.9023	0.4336	0.6094	0.6680	0.0234	0.1406
0.8438	0.0820	0.4922	0.9609	0.7852	0.7266
0.3750	0.2578	0.5508	0.3164	0.9023	0.4336
0.6094	0.6680	0.0234	0.1406	0.8438	0.0820
0.4922	0.9609	0.7852	0.7266	0.3750	0.2578
0.5508	0.3164	0.9023	0.4336	0.6094	0.6680
0.0234	0.1406	0.8438	0.0820		

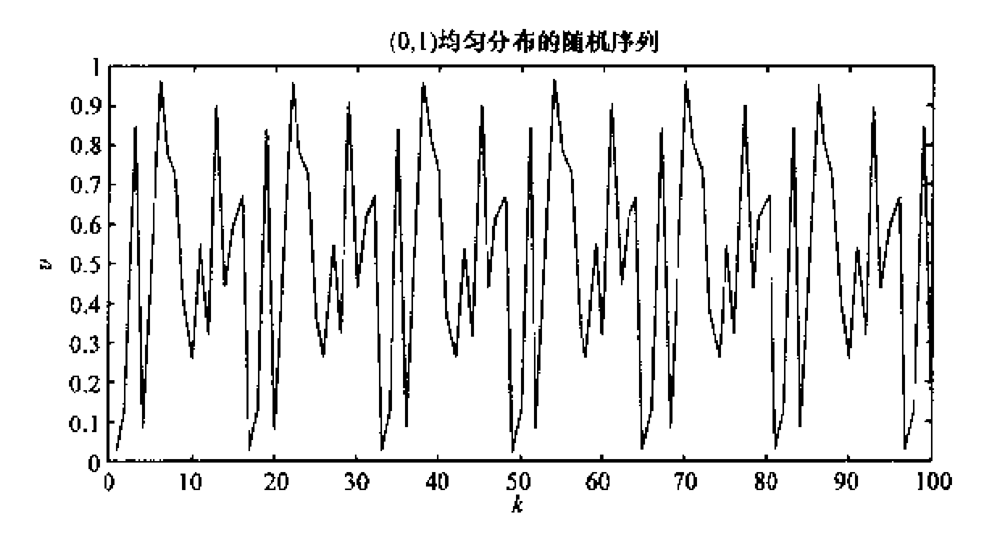


图 2.5 采用MATLAB产生的(0,1)均匀分布的随机序列图

3) 白噪声产生举例

例 2.2 利用乘同余法,仍选 $R=2,A=6,k=8,M=2^k=256$,递推100次,采用MATLAB的仿真语言编程,产生(-1,1)均匀分布的白噪声。

解 ① 只要将产生的(0,1)均匀分布的随机序列的程序稍加改动,将产生(0,1)均匀分布的随机数统统减去0.5,相当于将原随机序列图的横坐标向上平移0.5,原随机序列变成了(-0.5,0.5)的白噪声,然后乘以存储器 f 中预置的系数,便可得到任意幅值的白噪声,这里取 $f=2$,得到了产生(-1,1)均匀分布的白噪声的程序如下(附光盘上的该程序:FLch2sjxleg2.m):

```
A=6;  x0=1;    f=2;    N=100;  %初始化;
x0=1; M=255;
```

```
for k=1: N                   %乘同余法递推 100 次;
    x2=A*x0;                 %分别用 x2 和 x0 表示 x_i 和 x_{i-1};
    x1=mod(x2,M);            %取 x2 存储器的数除以 M 的余数放 x1 中;
    v1=x1/256;               %将 x1 存储器中的数除以 256 得到小于 1 的随机数放 v1 中;
    v(:,k)=(v1-0.5)*f;       %将 v1 中的数(ξ_i)减去 0.5 再乘以存储器 f 中的系数,
                             %存放在矩阵存储器 v 的第 k 列中;
      x0=x1;                 % x_{i-1}= x1,x1 中存放第 i 时刻的余数;
      v0=v1;     end         %递推 100 次结束;
v2=v                         %该语句末无';',实现矩阵存储器 v 中随机数放在 v2 中,
                             %且可直接显示在 MATLAB 的 window 中;
k1=k;
%grapher                     %绘图;
k=1:k1;
plot(k,v,k,v,'r');
xlabel('k'), ylabel('v');tktle('(-1,+1)均匀分布的白噪声')
```

② 程序运行结果如图 2.6 所示。

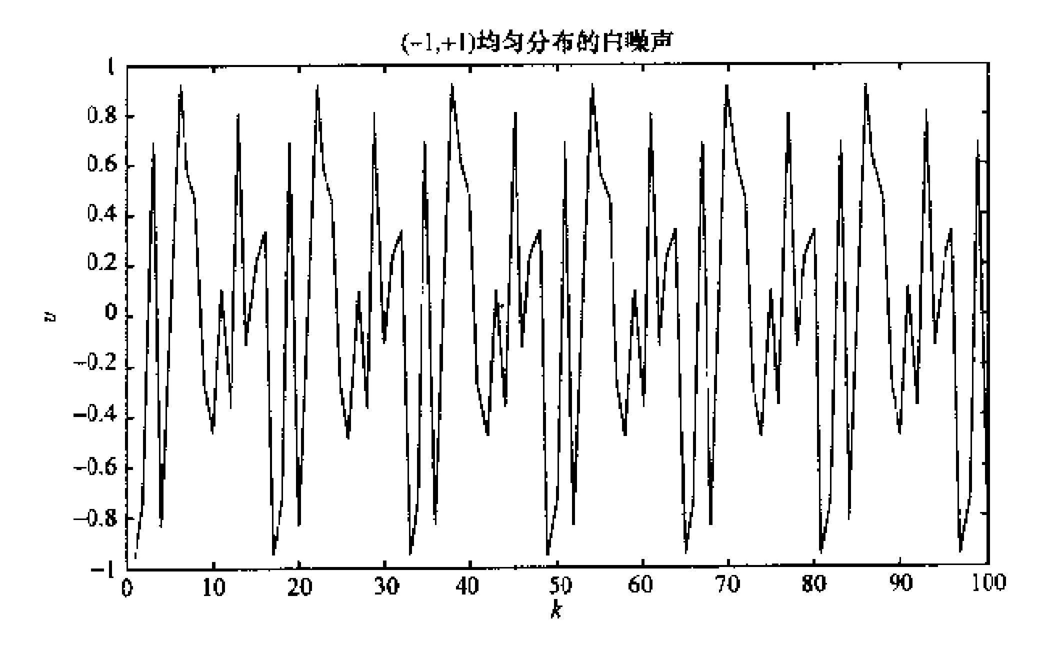


图 2.6　采用 MATLAB 产生的(-1,+1)均匀分布的白噪声序列

③ 产生的(-1,1)均匀分布的白噪声序列。

在程序运行结束后,产生的(-1,1)均匀分布的白噪声序列,直接从 MATLAB 的 window 界面中拷贝出来如下(v_2 中每行存 6 个随机数 $v_2(j,k)$,$j=1,2,\cdots,13$;$k=1,2,\cdots,6$):

```
v2 =

   -0.9531   -0.7188    0.6875   -0.8359   -0.0156    0.9219
    0.5703    0.4531   -0.2500   -0.4844    0.1016   -0.3672
```

0.8047	−0.1328	0.2188	0.3359	−0.9531	−0.7188
0.6875	−0.8359	−0.0156	0.9219	0.5703	0.4531
−0.2500	−0.4844	0.1016	−0.3672	0.8047	−0.1328
0.2188	0.3359	−0.9531	−0.7188	0.6875	−0.8359
−0.0156	0.9219	0.5703	0.4531	−0.2500	−0.4844
0.1016	−0.3672	0.8047	−0.1328	0.2188	0.3359
−0.9531	−0.7188	0.6875	−0.8359	−0.0156	0.9219
0.5703	0.4531	−0.2500	−0.4844	0.1016	−0.3672
0.8047	−0.1328	0.2188	0.3359	−0.9531	−0.7188
0.6875	−0.8359	−0.0156	0.9219	0.5703	0.4531
−0.2500	−0.4844	0.1016	−0.3672	0.8047	−0.1328
0.2188	0.3359	−0.9531	−0.7188	0.6875	−0.8359
−0.0156	0.9219	0.5703	0.4531	−0.2500	−0.4844
0.1016	−0.3672	0.8047	−0.1328	0.2188	0.3359
−0.9531	−0.7188	0.6875	−0.8359		

显然,只要在例 2.2 程序的初始化部分中给 $N=300, f=6$,运行程序就可以得到如图 2.3 所示的$(-3,3)$的白噪声过程。

当然,正态分布的随机数是我们最关心的一种。在已产生的(0,1)均匀分布的随机序列的基础上,可以推出正态分布的随机数表达式如式(2.33)所示。

$$\eta = \mu_\eta + \sigma_\eta \frac{\sum_{i=1}^{N} \xi_i - N/2}{\sqrt{N/12}} \tag{2.33}$$

式中,η 为(μ_η, σ^2)正态分布的随机数。当 ξ_i 为(0,1)均匀分布的随机数时,则

$$\mu_\eta = E\{\xi_i\} = \int_0^1 \xi_i p(\xi_i) \mathrm{d}\xi_i = 1/2 \tag{2.34}$$

$$\sigma_\eta = \sqrt{\mathrm{var}\{\xi_i\}} = \left[\int_0^1 (\xi_i - \mu_\eta) p(\xi_i) \mathrm{d}\xi_i\right]^{\frac{1}{2}} = [1/12]^{\frac{1}{2}} \tag{2.35}$$

式中,$E\{\xi_i\}$是 ξ_i 的均值;$\mathrm{var}\{\xi_i\}$是 ξ_i 的方差;当 $N=12$ 时,上式可简化为

$$\eta = \mu_\eta + \sigma_\eta\left(\sum_{i=1}^{N} \xi_i - 6\right) \tag{2.36}$$

式(2.36)为统计特性较理想的正态分布的随机数表达式。

2.4.3　伪随机信号产生及 MATLAB 仿真举例

1）基本概念

从辨识的概念可知，若系统的模型结构正确，系统模型的辨识精度直接通过 Fisher 信息函数矩阵依赖于输入来确定。因此，合理的选择系统辨识的输入信号是保证辨识精度的重要环节。前面已经讨论了白噪声的产生。但工业设备，如阀门、铲车、提升机等不可能按白噪声的变化动作。而线性移位寄存器序列（M 序列）是一种很好模拟工业设备动态运行中的辨识输入信号，M 序列具有接近白噪声的性质，因此称为伪随机信号。

2）用移位寄存器产生 M 序列

例 2.3　四级移位寄存器产生 M 序列的结构如图 2.7 所示。

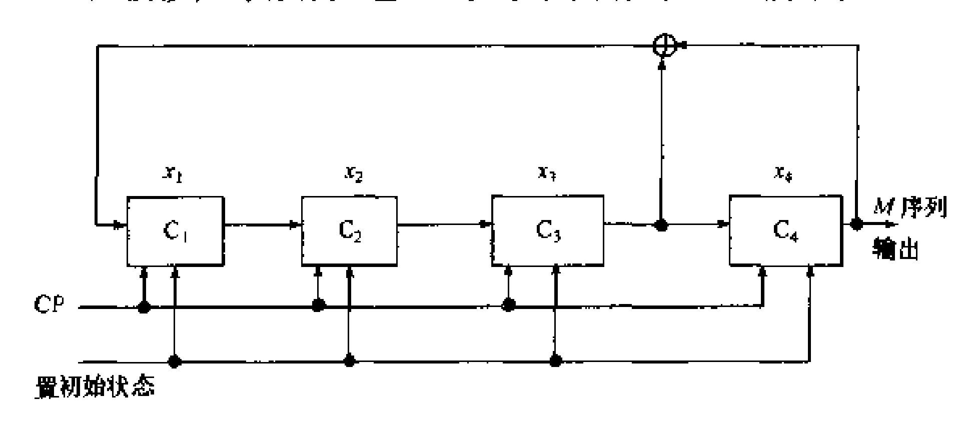


图 2.7　四级移位寄存器产生 M 序列的结构图

图中，⊕表示模 2 和（异或）运算，C_1、C_2、C_3 和 C_4 表示四个移位寄存器，x_1、x_2、x_3 和 x_4 分别表示各移位寄存器的输出，x_4 的输出为产生的 M 序列。四级移位寄存器的连接方式可用(2.37)表示。

$$\begin{cases} x_2(k+1) = x_1(k) \\ x_3(k+1) = x_2(k) \\ x_4(k+1) = x_3(k) \\ x_1(k+1) = x_3(k) \oplus x_4(k) \end{cases} \tag{2.37}$$

从而可见，x_1 第$(k+1)$时刻的状态由 x_3 和 x_4 第 k 时刻状态的异或运算结果决定；x_2 第$(k+1)$时刻的状态由 x_1 第 k 时刻状态决定；x_3 第$(k+1)$时刻的状态由 x_2 第 k 时刻状态决定；x_4 第$(k+1)$时刻的状态由 x_3 第 k 时刻状态决定。设置初始状态为 1010，在移位脉冲 CP 作用下，寄存器各级状态的变化如表 2.1。

表 2.1　四级移位寄存器产生 M 序列的工作状态

CP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
x_1	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0
x_2	0	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0
x_3	1	0	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
x_4	0	1	0	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1
$x_3 \oplus x_4$	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0

从表 2.1 可知 M 序列为：

x_4	0	1	0	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1

重复

该 M 序列可表示为

$$\{M(k)\} = s^2 \oplus s^4 \oplus s^5 \oplus s^6 \oplus s^7 \oplus s^{11} \oplus s^{14} \oplus s^{15} \oplus s^2 \oplus s^4 \oplus s^5 \oplus \cdots \tag{2.38}$$

式中，s^k 表示状态在第 k 位为'1'；$\oplus$表示模 2 和(异或)。该 M 序列有以下特点：

(1) M 序列的循环长度为 $N_p = 2^N - 1 = 15$，其中 N 为寄存器的个数；

(2) 逻辑为 1 的次数为 $\frac{2^N}{2} = 2^{N-1} = (N_p + 1)/2$；

(3) 逻辑为 0 的次数为 $\frac{2^N}{2} - 1 = 2^{N-1} - 1 = (N_p - 1)/2$；

(4) "游程"的特点。对四级移位寄存器产生 M 序列而言，每周期有 15 位(bit)。状态连续出现的段称为"游程"，该 15 个 bit 分为 8 段。其中'0'游程和'1'游程各有 4 个；长度为 1 个 bit 的有 4 个，占总段数的1/2；长度为 2 个 bit 的有 2 个，占总段数的1/4；长度为 3 个 bit 的有 1 个(逻辑 0)，占总段数的1/8；长度为 4 个 bit 的有 1 个(逻辑 1)，占总段数的1/8。

3) 用 MATLAB 软件实现移位寄存器产生 M 序列

例 2.4　现仍以四级移位寄存器产生 M 序列为例，在实际中，常把 M 序列的逻辑'0'和逻辑'1'变换成'a'和'$-a$'的序列，这里取 $a=1$，用软件实现。

解　① 编程如下(附光盘上的该程序：FLch2sjxleg3.m)：

```
X1=1;X2=0;X3=1;X4=0;   %移位寄存器输入 Xi 初态(0101),
m=60;                  %置 M 序列总长度 m 值;
for i=1:m              %开始循环
  Y4=X4;  Y3=X3;  Y2=X2;    Y1=X1;  % Yi 为移位寄存器各级输出,
                                    %在移位之前先将各自的输入传给输出;
  X4=Y3;  X3=Y2;    X2=Y1;          %实现移位寄存器的连接方式;
```

```
  X1 = xor(Y3,Y4);   %异或运算,实现 x1(k+1)=x3(k)⊕x4(k);
  if  Y4==0          %将输出'0'态转换成'-1'态;
  U(i)= -1;
  else
  U(i)=Y4;
end              %转换结束;
end              %60 次循环结束;
  M=U
  %绘图
  i1=i
  k=1:1:i1;
  plot(k,U,k,U,'rx')
  xlabel('k')
  ylabel('M 序列 ')
  title('移位寄存器产生的 M 序列 ')
```

② 程序运行结果如图 2.8 所示。

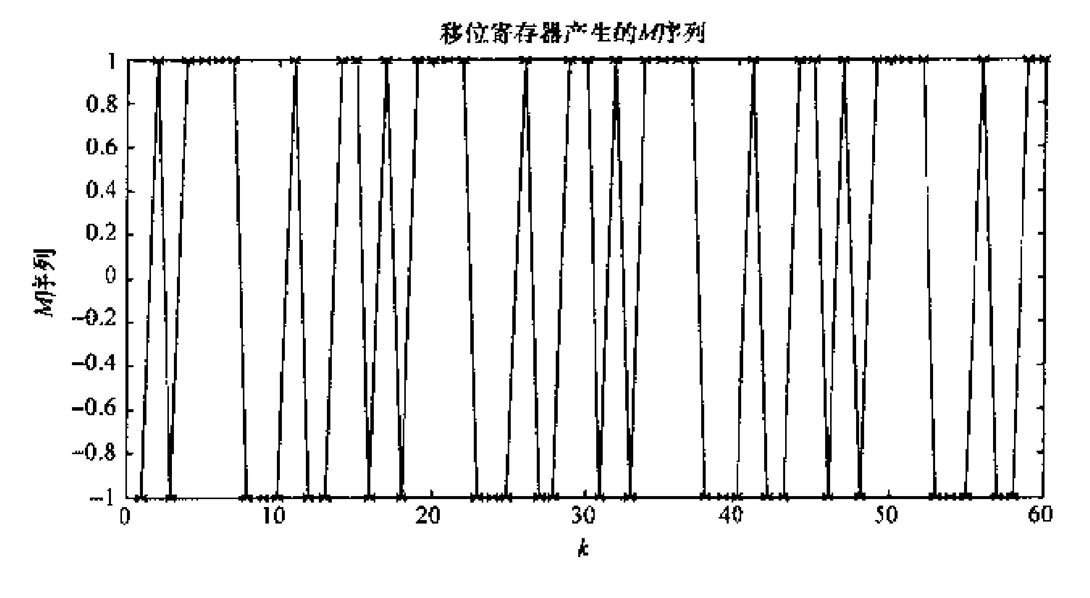


图 2.8 软件实现的移位寄存器产生的 M 序列图

③ 用软件实现的移位寄存器产生的 M 序列,在程序运行结束后,产生的 $(-1,1)$ M 序列,直接从 MATLAB 的 window 界面中拷贝出来如下(M 中每行存 10 个数($k=1\sim10$),在程序中置 M 序列总长度值 $m=60$,所以共 60 个数,$M(j,k),j=1,2,\cdots,6;k=1,2,\cdots,10$):

M=

-1	1	-1	1	1	1	1	-1	-1	-1
1	-1	-1	1	1	-1	1	-1	1	1
1	1	-1	-1	-1	1	-1	-1	1	1

-1	1	-1	1	1	1	1	-1	-1	-1
1	-1	-1	1	1	-1	1	-1	1	1
1	1	-1	-1	-1	1	-1	-1	1	1

在实际中,可根据所需 M 序列的周期长度的不同,选择移位寄存器的级数 N,只要适当的连接其结构,便可以得到满意的 N_p。在其结构中,决定 x_1 第 $k+1$ 时刻状态的第 k 时刻状态异或运算结果是关键。下面将常用的 N 级产生 M 序列移位寄存器连接方式列于表 2.2。

表 2.2 N 级产生 M 序列移位寄存器的连接方式

周期长度 $N_p=2^N-1$	状态异或运算	寄存器的级数 N
7	$x_1(k+1)=x_2(k)\oplus x_3(k)$	$N=3$
15	$x_1(k+1)=x_3(k)\oplus x_4(k)$	$N=4$
31	$x_1(k+1)=x_3(k)\oplus x_5(k)$	$N=5$
63	$x_1(k+1)=x_5(k)\oplus x_6(k)$	$N=6$
127	$x_1(k+1)=x_4(k)\oplus x_7(k)$	$N=7$
255	$x_1(k+1)=x_2(k)\oplus x_3(k)\oplus x_4(k)\oplus x_8(k)$	$N=8$
511	$x_1(k+1)=x_5(k)\oplus x_9(k)$	$N=9$
1023	$x_1(k+1)=x_7(k)\oplus x_{10}(k)$	$N=10$
2047	$x_1(k+1)=x_9(k)\oplus x_{11}(k)$	$N=11$

任何 M 序列均具有移位相加的性质。除了上述用软件直接实现实用的 N 级移位寄存器 M 序列外,还可以将 N 级移位寄存器 M 序列(如表 2.1)各级移位寄存器的状态任意组合成实用的 M 序列,如式(2.39)所示。

$$IM(k) = bs^2 \oplus x_1(k) \oplus x_4(k) \tag{2.39}$$

式中,s^2 为 01 状态,系数 b 和周期 N_p 的循环次数有关,可表示为

$$b = \begin{cases} 1, & k = iN_p, \quad i = 1,3,5,\cdots \\ 0, & k = iN_p, \quad i = 2,4,6,\cdots \end{cases} \tag{2.40}$$

$IM(k)$ 为由状态 01、寄存器 C_1 的输出 $x_1(k)$ 和寄存器 C_4 的输出 $x_4(k)$ 的在每个采样点上的状态模 2 和(异或)运算形成的序列。显然,对于式(2.39)也可以采用编程用软件来实现。

2.5 古典辨识方法

古典(经典)辨识方法(亦称为非参数辨识),在古典控制理论中,线性系统的动

态特性通常用传递函数 $G(s)$、频率响应 $G(j\omega)$、脉冲响应 $g(t)$ 或阶跃响应 $h(t)$ 来表示，后三种为非参数模型，其表现形式是时间或频率为自变量的实验曲线。对于系统施加不同的特定的实验信号，测定出相应的输出，可以求得这些非参数模型。经过适当的数学处理，它们又可以转变成参数模型，即传递函数。

获取非参数模型的主要方法有：阶跃响应法、脉冲响应法、频率响应法及相关分析法等。下面先介绍系统辨识的持续激励信号自相关函数。

2.5.1　*M* 序列自相关函数

系统辨识中，常把移位寄存器产生的 M 序列的逻辑 0 和逻辑 1 变换成 a 和 $-a$ 的序列作为系统的激励信号。在 M 序列表 2.1 中若从 $cp=4$(时钟的第 4 周期)算起(或者将 4 级移位寄存器的初始状态置成 1111)，将 4 级移位寄存器输出的逻辑 0 和逻辑 1 变换成 a 和 $-a$ 的序列，如图 2.9 所示。

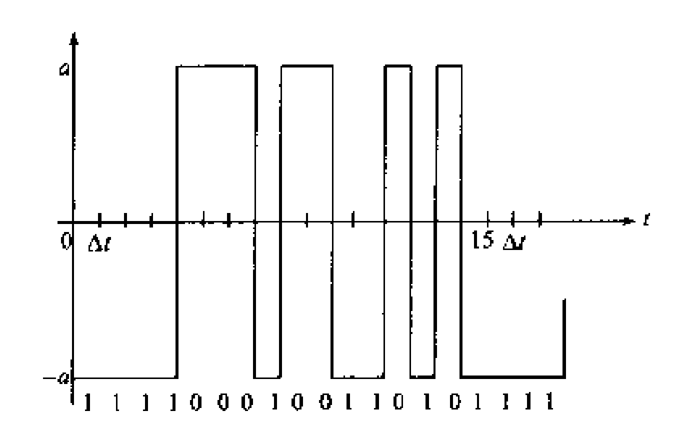


图 2.9　4 级 M 序列变换成幅值为 a 和 $-a$ 的序列

这种变换关系可表达为

$$M(i)=a(1-2x_i) \tag{2.41}$$

或

$$M(i)=a\cdot\exp[\mathrm{j}\pi x_i],\quad \mathrm{j}=\sqrt{-1} \tag{2.42}$$

式中，x_i 是取 0 或 1 的 M 序列元素；$M(i)$是取 a 或 $-a$ 的 M 序列元素。显然，在这种转换关系下，$\{M(i)\}$ 的乘法群 $(M(k)M(k+j))$ 和 $\{x_i\}$ 的加法群 $(x_k\oplus x_{k+j})$具有同构关系(即两者在相同初始状态下运算结果相同)。

根据自相关函数的定义，M 序列的自相关函数可以按下式计算

$$R_M(\tau)=\frac{1}{N_p\Delta t}\int_0^{N_p\Delta t}M(t)M(t+\tau)\mathrm{d}t \tag{2.43}$$

式中，$M(t)$是幅值为 a 或 $-a$ 的 M 序列；Δt 是位移脉冲周期(时钟周期)，上式的离散形式写成

$$R_M(\tau) = \frac{1}{N_p}\sum_{k=0}^{N_p-1} M(k)M(k+\tau) \tag{2.44}$$

根据上述同构关系式(2.41),显然,当 $\tau=0$ 时,$R_M(\tau)=a^2$;当 $\tau=j\Delta t$,$j(1\leqslant j\leqslant N_p-1)$为整数时,则有

$$R_M(\tau) = \frac{1}{N_p}\sum_{k=0}^{N_p-1} M(k)M(k+j) = -\frac{a^2}{N_p} \tag{2.45}$$

当 $-\Delta t\leqslant\tau\leqslant\Delta t$ 时

$$R_M(\tau) = a^2 - \frac{2|\tau|a^2}{N_p\Delta t}2^{p-1} = a^2\left[1-\frac{N_p+1}{N_p}\frac{|\tau|}{\Delta t}\right], |\tau|\leqslant\Delta t \tag{2.46}$$

因此,周期为 $N_p\Delta t$ 的 M 序列的自相关函数可归纳为

$$R_M(\tau) = \begin{cases} a^2\left[1-\dfrac{N_p+1}{N_p}\dfrac{|\tau|}{\Delta t}\right], & -\Delta t\leqslant\tau\leqslant\Delta t \\ \dfrac{a^2}{-N_p}, & \Delta t<\tau<(N_p-1)\Delta t \end{cases} \tag{2.47}$$

2.5.2 逆 M 序列

1) 逆 M 序列的产生

从谱密度角度分析,M 序列的谱密度是以$[2\pi a^2(N_p+1)/N_p][\sin(\omega\Delta t/2)/(\omega\Delta t/2)]$为包络线的线条谱[42],在 $\omega=0$ 时,周期为 N_p 的 M 序列的谱密度可描述为

$$S_w(\omega) = 2\pi a^2/N_p^2 \tag{2.48}$$

谱分析表明,M 序列含有直流成分,造成对辨识对象的净干扰。增加 N_p 可减小该直流成分。逆 M 序列可以达到这个目的。它是一种比 M 序列更理想的伪随机码序列。在 M 序列基础上生成逆 M 序列的方法很多,并且很容易实现。下面举例说明。

例 2.5 在表 2.2 中,4 级移位寄存器产生的 M 序列,初始状态为"1111",$N_p=15$,取 M 序列,即 $x_4(k)$的状态;$S(k)$是周期为 2bit,元素取 0 或 1 的方波序列;可用(2.49)生成逆 M 序列,并将产生过程列于表 2.3。

表 2.3 M 序列与方波复合生成逆 M 序列{$IM(k)$}过程

$M(k)$	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1	…
$S(k)$	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	…
⊕																				
$IM(k)$	0	1	0	1	1	0	1	1	1	0	0	1	1	1	1	1	0	1	0	…

$$\{IM(k)\} = \{M(k) \oplus S(k)\} \tag{2.49}$$

可见,该逆 M 序列$\{IM(k)\}$是 M 序列$\{M(k)\}$序列和方波$\{S(k)\}$在每个位上的模 2 和运算结果。但该逆 M 序列$\{IM(k)\}$的周期为 $2N_p=30$。

例 2.6　在 4 级移位寄存器产生的 M 序列表 2.1 中,$N_p=15$,从时钟 $cp=4$ 算起,分别取 M 序列的 $x_4(k)$和 $x_1(k)$序列,再利用(2.39)式 $IM(k)=bs^2\oplus x_1(k)\oplus x_4(k)$生成逆 M 序列。根据(2.40)分析,$s^2=01$,$b=iN_p$,$i=1,3,5,\cdots$,即 bs^2 项只有在第奇数个循环周期上的第 2 位出现,即 $k=2,32,\cdots$时为 1,其余位全为 0。将产生过程列于表 2.4。

表 2.4　$IM(k)=bs^2\oplus x_1(k)\oplus x_4(k)$生成逆 M 序列过程

k	1	2	3	4	5	6	7	8	…	16	17	18	19	20	…	31	32	33	…
x_1	1	0	0	0	1	0	0	1	…	1	0	0	0	1	…	1	0	0	…
x_4	1	1	1	1	0	0	0	1	…	1	1	1	1	0	…	1	1	1	…
bs^2	0	1	0	0	0	0	0	0	…	0	0	0	0	0	…	0	1	0	…
$\oplus$																			
IM	0	0	1	1	1	0	0	0	…	0	1	1	1	1	…	0	0	1	…

上两例生成的逆 M 序列均使 M 序列的周期增大到 $2N_p$,从而使其直流成分大大减小,若将逆 M 序列的逻辑 0 和逻辑 1 变换成 a 和 $-a$ 的序列,可以作为系统辨识中一种理想的激励信号。因此,逆 M 序列在辨识中应用广泛。

2) 逆 M 序列的性质

由此不难看出,逆 M 序列有如下性质

(1) 逆 M 序列$\{IM(k)\}$的周期 N_p^{I} 是原序列$\{M(k)\}$周期的两倍,即

$$N_p^{\mathrm{I}} = 2N_p \tag{2.50}$$

(2) 如果将逆 M 序列$\{IM(k)\}$的逻辑状态 0 换成$+a$,逻辑状态 1 换成$-a$,则在该序列中,$+a$ 和$-a$ 两个电平出现的频率相等,因此,该序列在 $N_p^{\mathrm{I}}=2N_p$ 内的平均值为零,在这方面类似于白噪声。

(3) 逆 M 序列$\{IM(k)\}$和原序列是不相关的。

2.5.3　相关分析法频率响应辨识

1) 基本概念

在古典控制理论中,系统的频率响应 $G(\mathrm{j}\omega)$表示为

$$G(\mathrm{j}\omega) = \frac{Y(\mathrm{j}\omega)}{R(\mathrm{j}\omega)} = \mathrm{Re}(\omega) + \mathrm{Im}(\omega) \tag{2.51}$$

式中,$\mathrm{Re}(\omega)$、$\mathrm{Im}(\omega)$分别表示频率响应的实频特性和虚频特性;$U(\mathrm{j}\omega)$,$Y(\mathrm{j}\omega)$分

别表示系统输入信号和输出信号的傅里叶变换，可写成

$$U(\mathrm{j}\omega)=\int_{-\infty}^{\infty}u(t)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}t=\int_{-\infty}^{\infty}u(t)\cos\omega t\mathrm{d}t+\mathrm{j}\int_{-\infty}^{\infty}u(t)\sin\omega t\mathrm{d}t \tag{2.52}$$

$$Y(\mathrm{j}\omega)=\int_{-\infty}^{\infty}y(t)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}t=\int_{-\infty}^{\infty}y(t)\cos\omega t\mathrm{d}t+\mathrm{j}\int_{-\infty}^{\infty}y(t)\sin\omega t\mathrm{d}t \tag{2.53}$$

确定（线性定常）系统的稳态时间响应 $c(t)$、频率响应 $G(\mathrm{j}\omega)$ 和输入信号 $U(\mathrm{j}\omega)$ 之间的关系可描述为

$$c(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}C(\mathrm{j}\omega)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}\omega=\frac{1}{2\pi}\int_{-\infty}^{\infty}G(\mathrm{j}\omega)U(\mathrm{j}\omega)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}\omega \tag{2.54}$$

在生产实际中，频率响应用解析方法直接求解太繁琐，很难找到。于是用频率特性图示法。频率特性图示法主要有：极坐标图（或奈魁斯特（Nyquist）图）；对数坐标（或伯德（Bode）图）；另外，闭环频率特性图示法有：等 M 圆和等 N 圆，尼柯尔斯图。大家熟知，在任何一本《自动控制原理》书上都有一定的篇幅讨论频率特性图的绘制方法。换句话说，系统的频率响应利用古典控制理论频率特性图示法可以方便的得到。

最小相位系统的传递函数通常可以用一些基本环节描述为

$$G(s)=\frac{K\prod_{i=1}^{p}(T_{1i}s+1)\prod_{i=1}^{q}(T_{2i}^{2}s^{2}+2T_{2i}\xi_{2i}s+1)}{s^{n}\prod_{i=1}^{r}(T_{3i}s-1)\prod_{i=1}^{l}(T_{4i}^{2}s^{2}+2T_{4i}\xi_{4i}s+1)} \tag{2.55}$$

从而可见，如果用实验法测得了系统的频率响应：系统的 Nyquist 曲线或 Bode 图，可依据在不同频率下 Nyquist 曲线的实频特性 $\mathrm{Re}(\omega)$ 和虚频特性 $\mathrm{Im}(\omega)$ 直接写出系统的开环频率响应 $G(\mathrm{j}\omega)$，或依据在不同频率下 Bode 图的幅频特性 $L(\omega)$ 和相频特性 $\phi(\omega)$ 可直接写出系统的开环传递函数 $G(s)$。再将传递函数变成频率响应 $G(\mathrm{j}\omega)$，即

$$G(\mathrm{j}\omega)=G(s)\big|_{s=\mathrm{j}\omega} \tag{2.56}$$

2）相关分析法频率响应辨识

上述频率响应法只有在信噪比很高的情况下才有效。然而在工程实际中，系统输出中含有测量噪声如图 2.10 所示，获得的数据总是含有噪声，相关分析法可有效的解决频率响应辨识问题。图中，系统的输入、输出分别为 $u(t)$ 和 $y(t)$；测量输出和测量噪声分别为 $z(t)$ 和 $w(t)$。

相关函数的定义：

$$\begin{cases}R_{uz}(\tau)=\lim\limits_{T\to\infty}\dfrac{1}{T}\displaystyle\int_{0}^{T}z(t)u(t-\tau)\mathrm{d}t\\ R_{uu}(\tau)=\lim\limits_{T\to\infty}\dfrac{1}{T}\displaystyle\int_{0}^{T}u(t)u(t-\tau)\mathrm{d}t\end{cases} \tag{2.57}$$

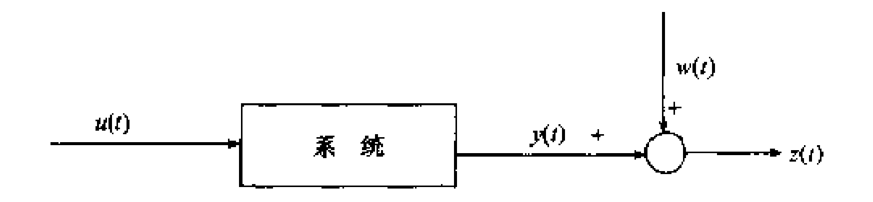


图 2.10　系统输出中含有测量噪声

其离散形式写成

$$\begin{cases} R_{uz}(\tau) = \dfrac{1}{T}\sum\limits_{t=0}^{T} z(t)u(t-\tau) \\ R_{uu} = \dfrac{1}{T}\sum\limits_{t=0}^{T} u(t)u(t-\tau) \end{cases} \tag{2.58}$$

式中，$R_{uu}(\tau)$指系统的输入的自相关函数；$R_{uz}(\tau)$指系统的输入和输出的互相关函数；互相关函数是时间为频率 ω 的函数，即 $T=2\pi/\omega$。

在图 2.10 中，设噪声 $w(t)$是均值为零的白噪声，其相关函数为 $R_w(\tau)=\sigma_w^2\delta(\tau)$。当输入信号为正弦波 $u(t)=A\sin\omega t$ 时，系统的输出稳态相应可表示成

$$z(t) = \sum_{k=1}^{\infty} B_k\sin(k\omega t+\theta_k) + w(t) \tag{2.59}$$

在时间 $T=2n\pi/\omega$(n 为整数)区间上，计算 $z(t)$与 $\sin\omega t$ 在$\tau=0$ 时的互相函数

$$\begin{aligned} z_s \triangleq R_{z(t)\sin\omega t}(0) &= \frac{1}{T}\int_0^T z(t)\sin\omega t\,\mathrm{d}t \\ &= \frac{B_1}{T}\int_0^T \sin(\omega t+\theta_1)\sin\omega t\,\mathrm{d}t + \sum_{k=2}^{\infty}\frac{B_k}{T}\int_0^T \sin(k\omega t+\theta_k)\sin\omega t\,\mathrm{d}t \\ &\quad + \frac{1}{T}\int_0^T w(t)\sin\omega t\,\mathrm{d}t \\ &\triangleq I_{s1}+I_{s2}+I_{s3} \end{aligned} \tag{2.60}$$

由于 $T=2n\pi/\omega$，利用三角函数的性质，可以推导出

$$I_{s1} = \frac{1}{2}B_1\cos\theta_1 \tag{2.61}$$

及

$$I_{s2} = 0 \tag{2.62}$$

当 T 充分大时，即 T 不再是 $2\pi/\omega$ 的整倍数，式(2.61)和式(2.62)仍近似成立。

在式(2.60)中，由于 I_{s3}是白噪声 $w(t)$的函数，因此 I_{s3}是随机量，显然由于 $w(t)$的均值 $E\{w(t)\}=0$，所以 $E\{I_{s3}\}=0$。考虑到随机过程的自相关函数 $R_w(t_1,t_2)$只与时间差(t_2-t_1)有关，而与 t_1 和 t_2 的值无关，因此，I_{s3}的自相关函数可写成

$$R_{I_{s3}}(\tau)=R_{I_{s3}}(t_2-t_1)=E\left\{\left[\frac{1}{T}\int_0^T w(t_1)\sin\omega t_1\mathrm{d}t_1\right]\left[\frac{1}{T}\int_0^T w(t_2)\sin\omega t_2\mathrm{d}t_2\right]\right\}$$
$$=\frac{1}{T^2}\int_0^T\sin\omega t_1\mathrm{d}t_1\int_0^T\sin\omega t_2E\{w(t_1)w(t_2)\}\mathrm{d}t_2$$
$$=\frac{1}{T^2}\int_0^T\sin\omega t_1\mathrm{d}t_1\int_0^T\sin\omega t_2\sigma_w^2\delta(t_2-t_1)\mathrm{d}t_2$$
$$=\frac{\sigma_w^2}{T^2}\int_0^T\sin^2\omega t_1\mathrm{d}t_1=\frac{\sigma_w^2}{2T}\left(1-\frac{\sin2\omega T}{2\omega T}\right)\tag{2.63}$$

式中,σ_w^2 为噪声 $w(t)$ 的方差,它可以写成的形式

$$\sigma_w^2(\tau)=\mathrm{var}\{w(t)\}=R_w(0)-E\{w(t)\}\tag{2.64}$$

在式(2.63)中,当 $T=2n\pi/\omega$ 时,则

$$R_{I_{s3}}(t_2-t_1)=\frac{\sigma_w^2}{2T}\tag{2.65}$$

显然,只要积分周期 T 充分大时,I_{s3}的自相关函数就趋于零。可见,I_{s3}相对于 I_{s1} 来说完全可以忽略,因此式(2.60)可简化为

$$z_s\triangleq R_{z(t)\sin\omega t}(0)=\frac{1}{2}B_1\cos\theta_1\tag{2.66}$$

依此类推,在时间 $T=2n\pi/\omega$(n 为整数)区间上,计算 $z(t)$与 $\cos\omega t$ 在 $\tau=0$ 时的互相函数

$$z_c\triangleq R_{z(t)\cos\omega t}(0)=\frac{1}{T}\int_0^T z(t)\cos\omega t\mathrm{d}t$$
$$=\frac{B_1}{T}\int_0^T\sin(\omega t+\theta_1)\cos\omega t\mathrm{d}t+\sum_{k=2}^{\infty}\frac{B_k}{T}\int_0^T\sin(k\omega t+\theta_k)\cos\omega t\mathrm{d}t$$
$$+\frac{1}{T}\int_0^T w(t)\cos\omega t\mathrm{d}t$$
$$\triangleq I_{c1}+I_{c2}+I_{c3}\tag{2.67}$$

同理,当积分周期 T 充分大时,可推导出

$$\begin{cases}I_{c1}=\dfrac{1}{2}B_1\sin\theta_1\\ I_{c2}=0\\ I_{c3}=0,\because E\{I_{c3}\}=0,R_{I_{c3}}(t_2-t_1)=\dfrac{\sigma_w^2}{T^2}\left(1-\dfrac{\sin2\omega T}{2\omega T}\right)\approx\dfrac{\sigma_w^2}{2T}\to0\end{cases}\tag{2.68}$$

$$z_c\triangleq R_{z(t)\cos\omega t}(0)=\frac{1}{2}B_1\sin\theta_1\tag{2.69}$$

因此，过程的频率响应估计可以写成

$$\begin{cases}\|\hat{G}(\mathrm{j}\omega)\| = \dfrac{B_1}{A} = \dfrac{2}{A}\sqrt{z_s^2 + z_c^2} \\ \angle \hat{G}(\mathrm{j}\omega) = \theta_1 = \arctan\dfrac{z_c}{z_s}\end{cases} \tag{2.70}$$

或

$$\begin{cases}\mathrm{Re}(\omega) = \dfrac{2z_c}{A} \\ \mathrm{Im}(\omega) = \dfrac{2z_s}{A}\end{cases} \tag{2.71}$$

式中，A 为系统输入信号的幅值。综合式(2.70)和(2.71)，系统的频率响应估计也可以写成

$$\begin{aligned} G(\mathrm{j}\omega) &= \|\hat{G}(\mathrm{j}\omega)\|\angle\hat{G}(\mathrm{j}\omega) = \frac{2}{A}\sqrt{z_s^2 + z_c^2}\arctan\frac{z_c}{z_s} \\ &= \mathrm{Re}(\omega) + \mathrm{Im}(\omega) = \frac{2z_c}{A} + \frac{2z_s}{A} \end{aligned} \tag{2.72}$$

从而可见，在系统输入正弦信号，即 $u(t)=\sin\omega t$ 时，$w(t)$ 为白噪声，系统的频率响应估计求法是：只要先分别按式(2.66)和(2.69)计算出互相关函数 z_s 和 z_c，然后按式(2.72)求出所需要形式的频率响应。

2.5.4 相关分析法脉冲响应辨识

1. 脉冲响应辨识的概念

脉冲响应是在理想脉冲输入作用下过程的输出响应。考虑到工程上时间输入为理想脉冲是不可能的，因此通常采用矩形脉冲输入。当矩形脉冲的宽度比系统的过渡时间小得多、且矩形脉冲的面积等于1时，过程的输出可近似为脉冲响应。

当系统存在噪声时，利用相关分析法辨识系统的脉冲响应可以得到较理想的效果。

利用相关分析法辨识系统的脉冲响应的基本原理如图2.11所示。图中，$u(t)$ 是均值为零的噪声，其相关函数为 $R_w(\tau)=\sigma_w^2\delta(\tau)$；$\tilde{z}(t)$ 是过程与模型的输出误差，它与过程输出 $z(t)$ 和模型输出 $\hat{z}(t)$ 的关系为

$$\tilde{z}(t) = z(t) - \hat{z}(t) \tag{2.73}$$

式中

$$\hat{z}(t) = \int_0^{\infty} g(\theta)u(t-\theta)\mathrm{d}\theta \tag{2.74}$$

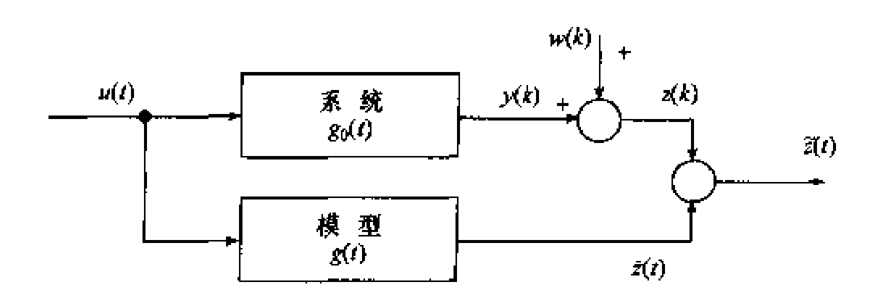


图 2.11　脉冲响应辨识示意图

考虑如下准则函数

$$J=\lim_{T\to\infty}\frac{1}{T}\int_0^T \tilde{z}^2(t)\mathrm{d}t \tag{2.75}$$

式中,T 为积分周期。现在的任务是根据系统的输入输出数据,确定模型的脉冲响应 $g(t)$,使误差准则函数 $J=\min$,即需要解决准则函数的极小化问题。综合式(2.73)~(2.75),准则函数为

$$J=\lim_{T\to\infty}\frac{1}{T}\int_0^T\left[z(t)-\int_0^\infty g(\theta)u(t-\theta)\mathrm{d}\theta\right]^2\mathrm{d}t \tag{2.76}$$

这是一个典型的变分问题。

设 $g(\theta)=\hat{g}(\theta)$时,准则函数 J 达到极小值,必要条件是

$$\lim_{\alpha\to 0}\frac{\partial J[\hat{g}(\theta)+\alpha g_\alpha(\theta)]}{\partial\alpha}=0 \tag{2.77}$$

式中,$g_\alpha(\theta)$是不为零的任意小波动函数;α 是标量。

考虑到式(2.76),式(2.77)展开为

$$\begin{aligned}&\lim_{\alpha\to 0}\frac{\partial J[\hat{g}(\theta)+\alpha g_\alpha(\theta)]}{\partial\alpha}\\&=\lim_{\alpha\to 0}\frac{\partial}{\partial\alpha}\left\{\lim_{T\to\infty}\frac{1}{T}\int_0^T\left[z(t)-\int_0^\infty\hat{g}(\theta)+\alpha g_\alpha(\theta)u(t-\theta)\mathrm{d}\theta\right]^2\mathrm{d}t\right\}\\&=\lim_{T\to\infty}\left(-\frac{2}{T}\right)\int_0^T\left\{\left[z(t)-\int_0^\infty\hat{g}(\theta)u(t-\theta)\mathrm{d}\theta\right]\int_0^\infty g_\alpha(\theta)u(t-\theta)\mathrm{d}\theta\right\}\mathrm{d}t=0\end{aligned} \tag{2.78}$$

为了不引起混乱,先将上式的中括号外$\int_0^\infty g_\alpha(\theta)u(t-\theta)\mathrm{d}\theta$ 的积分变量θ 换成τ,然后变换积分次序,则有

$$\int_0^\infty g_\alpha(\tau)\left\{\lim_{T\to\infty}\left(\frac{1}{T}\right)\int_0^T\left[z(t)-\int_0^\infty\hat{g}(\theta)u(t-\theta)\mathrm{d}\theta\right]u(t-\tau)\mathrm{d}t\right\}\mathrm{d}\tau=0 \tag{2.79}$$

由于 $g_\alpha(\tau)$是不为零的任意小波动函数,则有

$$\lim_{T\to\infty}\left(\frac{1}{T}\right)\int_0^T\left[z(t)-\int_0^\infty \hat{g}(\theta)u(t-\theta)\mathrm{d}\theta\right]u(t-\tau)\mathrm{d}t=0 \tag{2.80}$$

或

$$\begin{aligned}&\lim_{T\to\infty}\left(\frac{1}{T}\right)\int_0^T z(t)u(t-\tau)\mathrm{d}t\\&=\int_0^\infty \hat{g}(\theta)\left[\lim_{T\to\infty}\left(\frac{1}{T}\right)\int_0^T u(t-\tau)u(t-\theta)\mathrm{d}t\right]\mathrm{d}\theta\end{aligned} \tag{2.81}$$

若设系统的输入输出数据是平稳的各态遍历的随机过程，则上式的右侧中括号里的内容为输入的自相关函数，而上式的左边为输入输出的互相关函数，即

$$\begin{aligned}R_{uz}&=\lim_{T\to\infty}\left(\frac{1}{T}\right)\int_0^T z(t)u(t-\tau)\mathrm{d}t\\&=\int_0^\infty \hat{g}(t)R_u(t-\tau)\mathrm{d}t\end{aligned} \tag{2.82}$$

上式称为 Wiener-Hopf 方程。它是辨识过程脉冲响应的理论根据。如果输入信号是均值为零的白噪声，其自相关函数为

$$R_u(\tau)=\sigma_u^2\delta(\tau) \tag{2.83}$$

则式(2.82)变为

$$R_{uz}(\tau)=\int_0^\infty \hat{g}(t)\delta(\tau)\mathrm{d}t=\sigma_u^2\hat{g}(\tau) \tag{2.84}$$

从而可以得到

$$\hat{g}(\tau)=\frac{1}{\sigma_u^2}R_{uz}(\tau) \tag{2.85}$$

从而可见，只要计算出系统输入输出数据的互相关函数，就可以方便的求出脉冲响应的估计。

2. 用 M 序列作输入信号的响应估计的一般算法

由于 M 序列的统计特性近似于白噪声，根据 Wiener-Hopf 方程，M 序列响应为

$$R_{Mz}(\tau)\approx\int_0^{N_p\Delta t}\hat{g}(t)R_M(t-\tau)\mathrm{d}t \tag{2.86}$$

式中，N_p 是 M 序列的循环长度；Δt 是 M 序列移位脉冲的周期。当 N_p 充分大时，

$$R_{Mz}(\tau)\approx a^2\hat{g}(\tau) \tag{2.87}$$

式中，a 为 M 序列的幅值。可见输入信号用 M 序列和用白噪声结果是类似的、但此时脉冲响应只需计算一个周期的互相关函数 $R_{Mz}(\tau)$，大大的缩短了辨识的时间。可见，这种方法对脉冲响应的估计值的计算相对简单得多。

当数据采集的时间和 Δt 相等时，则式(2.86)的离散形式为

$$R_{Mz}(k) \approx \sum_{j=0}^{N_p-1} \hat{g}(j) R_M(k-j)\Delta t \tag{2.88}$$

M 序列的响应相关函数可写成

$$\begin{cases} R_{Mz}(k) = \dfrac{1}{N_p}\sum\limits_{i=0}^{N_p-1} M(i-k)z(i) \\ R_M(k) = \dfrac{1}{N_p}\sum\limits_{i=0}^{N_p-1} M(i-k)M(i) \end{cases} \tag{2.89}$$

从式(2.89)和(2.88)，可推导出 M 序列的自相关函数为

$$R_M(k) = \begin{cases} a^2, & k = 0, N_p, 2N_p, \cdots \\ -\dfrac{a^2}{N_p}, & k \neq 0, N_p, 2N_p, \cdots \end{cases} \tag{2.90}$$

系统输入输出的互相关函数为

$$R_{Mz}(k) = \frac{(N_p+1)a^2\Delta t}{N_p}\hat{g}(k) - \frac{a^2\Delta t}{N_p}\sum_{i=0}^{N_p-1}\hat{g}(i) \tag{2.91}$$

令

$$c = \frac{a^2\Delta t}{N_p}\sum_{i=0}^{N_p-1}\hat{g}(i) \tag{2.92}$$

对一个稳定的系统来说，c 是有界常数，而且很小(当 N_p 充分大时)，则有

$$\hat{g}(k) = \frac{N_p}{(N_p+1)a^2\Delta t}[R_{Mz}(k) + c] \tag{2.93}$$

式(2.93)是利用相关分析法辨识脉冲响应的一个重要公式。它描述了脉冲响应估计值与互相关函数之间的关系。如图 2.12 所示。

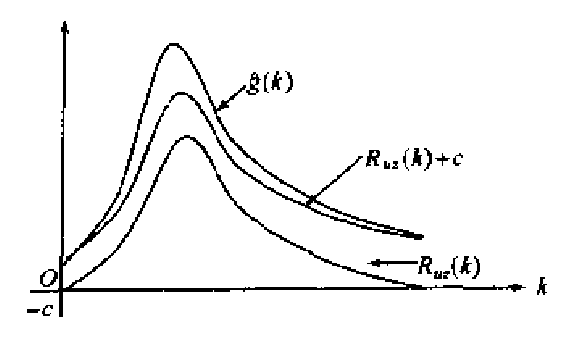


图 2.12　脉冲响应估计值与互相关函数的关系

对于一个稳定的系统来说，当 $k \to \infty$ 时，$\hat{g}(k) \to 0$，所以根据式(2.93)，有 $c = -R_{Mz}(\infty)$。在工程上，一般取 $c = -R_{Mz}(N_p - 1)$就已经足够满意了；有时甚至可以把 c 忽略不计。c 的取值并不影响脉冲响应的形态，而只决定脉冲响应在纵坐标上的位置。

3. 用 M 序列作输入信号的一次完成算法

根据式(2.88),当 $k=0,1,2,\cdots,N_p-1$ 时,得到 N_p 个线性联立方程

$$\begin{cases} R_M(0)\hat{g}(0)+R_M(-1)\hat{g}(1)+\cdots+R_M(-N_p+1)\hat{g}(N_p-1)=R_{Mz}(0)/\Delta t \\ R_M(1)\hat{g}(1)+R_M(0)\hat{g}(0)+\cdots+R_M(-N_p+2)\hat{g}(N_p-1)=R_{Mz}(1)/\Delta t \\ \qquad\vdots \\ R_M(N_p-1)\hat{g}(0)+R_M(N_p-2)\hat{g}(1)+\cdots+R_M(0)\hat{g}(N_p-1)=R_{Mz}(N_p-1)/\Delta t \end{cases} \tag{2.94}$$

将上式写成矩阵的形式

$$\boldsymbol{R}_M\hat{\boldsymbol{g}}=\boldsymbol{r}_{Mz}/\Delta t \tag{2.95}$$

式中

$$\begin{cases} \hat{\boldsymbol{g}}=[\hat{g}(0),\hat{g}(1),\cdots,\hat{g}(N_p-1)]^{\mathrm{T}} \\ \boldsymbol{R}_M=\begin{bmatrix} R_M(0) & R_M(-1) & \cdots & R_M(-N_p+1) \\ R_M(1) & R_M(0) & \cdots & R_M(-N_p+2) \\ \vdots & \vdots & & \vdots \\ R_M(N_p-1) & R_M(N_p-2) & \cdots & R_M(0) \end{bmatrix} \\ \qquad=-\dfrac{a^2}{N_p}\begin{bmatrix} -N_p & 1 & \cdots & 1 \\ 1 & -N_p & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & -N_p \end{bmatrix} \\ \boldsymbol{r}_{Mz}=[R_{Mz}(0),R_{Mz}(1),\cdots,R_{Mz}(N_p-1)]^{\mathrm{T}} \end{cases} \tag{2.96}$$

式中,$\boldsymbol{r}_{Mz}$为在 M 序列作用下系统输入输出的相关函数矩阵。由式(2.95),得

$$\hat{\boldsymbol{g}}=\boldsymbol{R}_M^{-1}\boldsymbol{r}_{Mz}/\Delta t \tag{2.97}$$

式中

$$\boldsymbol{R}_M^{-1}=-\frac{N_p}{a^2}\begin{bmatrix} -N_p & 1 & \cdots & 1 \\ 1 & -N_p & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & -N_p \end{bmatrix}^{-1} \tag{2.98}$$

由于

$$
\begin{bmatrix} -N_p & 1 & \cdots & 1 \\ 1 & -N_p & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & -N_p \end{bmatrix}
\begin{bmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 2 \end{bmatrix}
$$

$$
= \begin{bmatrix} -(N_p+1) & 0 & \cdots & 0 \\ 0 & -(N_p+1) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & -(N_p+1) \end{bmatrix} = -(N_p+1)\boldsymbol{I} \tag{2.99}
$$

因此

$$
\boldsymbol{R}_M^{-1} = \frac{N_p}{(N_p+1)a^2}\begin{bmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 2 \end{bmatrix} \tag{2.100}
$$

根据式(2.89),互相关函数阵又可以写成

$$
\boldsymbol{r}_{Mz} = \frac{1}{N_p}\boldsymbol{Mz} \tag{2.101}
$$

式中

$$
\begin{cases} \boldsymbol{z} = [z(0), z(1), \cdots, z(N_p-1)]^{\mathrm{T}} \\ \boldsymbol{M} = \begin{bmatrix} M(0) & M(1) & \cdots & M(N_p-1) \\ M(-1) & M(0) & \cdots & M(N_p-2) \\ \vdots & \vdots & & \vdots \\ M(-N_p+1) & M(-N_p+2) & \cdots & M(0) \end{bmatrix} \end{cases} \tag{2.102}
$$

综合式(2.97)~(2.101),得

$$
\hat{\boldsymbol{g}} = \frac{1}{N_p\Delta t}\boldsymbol{R}_M^{-1}\boldsymbol{Mz} = \frac{1}{(N_p+1)a^2\Delta t}\begin{bmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 2 \end{bmatrix}\boldsymbol{Mz} \tag{2.103}
$$

式中,常数矩阵为 N_p+1 维方阵。式(2.103)为 M 序列输入脉冲响应的一次完成算法。

4. 用 M 序列作输入信号的递推算法

为了在线辨识系统的脉冲响应,将第 i 时刻的互相关函数 $R_{Mz}^{(i)}(k)$ 写成递推的形式

$$R_{Mz}^{(i)}(k)=\frac{i}{i+1}R_{Mz}^{(i-1)}(k)+\frac{1}{i+1}M(i-k)z(i) \tag{2.104}$$

式中，$R_{Mz}^{(i-1)}(k)$为第$(i-1)$时刻的互相关函数。

根据式(2.97)和(2.104)，第 i 时刻的脉冲响应估计值的递推形式为

$$\hat{g}^{(i)}=\frac{i}{i+1}\hat{g}^{(i-1)}+\frac{1}{(i+1)\Delta t}\boldsymbol{R}_M^{-1}\boldsymbol{m}(i)z(i) \tag{2.105}$$

式中，$\hat{g}^{(i-1)}$表示采样至第$(i-1)$时刻的脉冲响应估计值，且

$$\boldsymbol{m}(i)=[M(i),M(i-1),\cdots,M(i-N_p+1)]^{\mathrm{T}} \tag{2.106}$$

5. 用 M 序列作输入信号辨识脉冲响应的步骤

(1) 预估被辨识系统的过渡过程时间 T_s 和系统的最高工作频率 $f_{\max}$。T_s 和 $f_{\max}$是选择 M 序列参数的依据。

(2) 在系统辨识中，数据必须尽可能多的包含系统动态特性的信息。这和输入信号的选择有很大的关系。因此，实验之前要精心地选择 M 序列的参数。当系统的频率特性接近低通滤波特性时，M 序列的参数N_p 和 Δt 应满足下述条件

$$\begin{cases}\dfrac{1}{3\Delta t}\geqslant f_{\max} & (2.107)\\ (N_p-1)\Delta t>T_s & (2.108)\end{cases}$$

式(2.107)表明，序列的频带必须覆盖系统的频带，这样才能充分激励系统的所有模态。式(2.108)表明，M 序列的循环周期必须大于系统的过渡过程时间，以保证时间大于 $N_p\Delta t$ 后，脉冲响应衰减接近于零。另外，M 序列的幅度不能选择过大，以免系统进入非线性区或影响系统生产；但也不能过小，以保证一定的信噪比。

(3) 采集数据时要注意，当 M 序列刚加上时，由于非零初始条件的作用，系统的输出在一段时间内时是非平稳的。为了保证辨识精度，要避开这段非平稳过程，一般可以从第二个循环周期开始采集数据。

(4) 数据要除去直流分量，有条件的还要进行滤波处理。

(5) 计算互相关函数 $R_{Mz}(k)$如果数据较多，要用快速傅里叶变换(FFT)方法计算相关函数，以减少计算量。

(6) 取补偿 $c=-R_{Mz}(N_p-1)$。

(7) 利用式(2.93)计算脉冲响应估计量$\hat{g}(k)$。

(8) 若采用 M 序列作输入信号，辨识步骤同上。

2.5.5 相关分析法脉冲响应应用

例 2.7　某炼油厂常压加热锅炉炉膛温度由汽动燃料调节阀膜头压力控制，

如图 2.13 所示。利用相关分析法辨识汽动调节膜头压力到炉膛温度通道的脉冲响应。

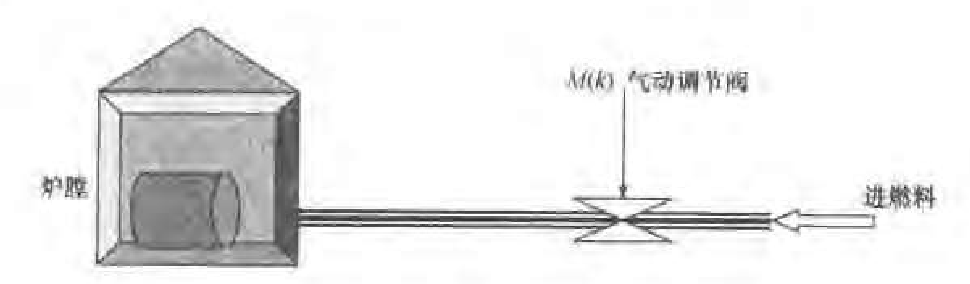


图 2.13　压力干扰加热炉

解

(1) 预估炉子的过渡过程时间 T_s 不大于 50min，最高工作频率 f_{max} 低于 0.0012Hz。

(2) 选择 M 序列参数。根据式(2.107)，取 $\Delta t = 4$min，使 $1/3\Delta t = 0.001\,39$Hz $>0.001\,2$Hz；根据式(2.108)，确定 $N_p = 15$，保证 $(N_p - 1)\Delta t = 56$min；且 M 序列按表 2.5 取值，另外，根据运行经验，气动调节阀膜头压力扰动幅度取 0.03 kg/cm^2，可保证对象不进入非线性区，并有明显的输出响应。

表 2.5　压力扰动记录

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$M(k)$	+	−	−	−	−	+	+	+	−	+	+	−	−	+	−

表中，"+"代表 0.03kg/cm^2；"−"代表 −0.03kg/cm^2。

(3) 为了获得温度响应的平稳过程，要在压力扰动加入一个循环周期开始记录数据。各个时刻的系统输出观测值如表 2.6 所示。

(4) 利用表 2.5 和表 2.6 的数据，计算互相关函数 $R_{Mz}(k)$。

(5) 根据式(2.93)和以上条件，可推出计算脉冲响应估计值的表达式为

$$\hat{g}(k) = \frac{15}{4 \times 0.03^2 \times 16}[R_{Mz}(k) + c] = 260.4[R_{Mz}(k) + c] \tag{2.109}$$

式中，互相关函数 $R_{Mz}(k) = \sum_{i=0}^{14} M(i-k)z(k)$。

式(2.109)中，取 $c = -R_{Mz}(14) = 0.0088$。对应的互相关函数 $R_{Mz}(k)$ 和脉冲响应估计值计算结果如表 2.7 所示。如果使用比较充足的数据(N_p 再取大些)，辨识精度还可以进一步提高。

表 2.6　各个时刻的系统输出观测值

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$z(k)$	1.82	1.82	2.03	2.03	1.03	0.68	0.52	0.86	1.78	2.50	2.50	2.32	3.28	2.82	2.04

表 2.7　互相关函数 $R_{Mz}(k)$和脉冲响应估计值

k	0	1	2	3	4	5	6
$R_{Mz}(k)$	-0.93×10^{-2}	-0.87×10^{-2}	0.64×10^{-2}	1.09×10^{-2}	0.76×10^{-2}	0.26×10^{-2}	-0.16×10^{-2}
$\hat{g}(k)$	−0.13	0.03	3.96	5.13	4.27	2.97	1.87

k	7	8	9	10	11	12	13	14
$R_{Mz}(k)$	-0.19×10^{-2}	-039×10^{-2}	-0.77×10^{-2}	-0.94×10^{-2}	-0.85×10^{-2}	-0.71×10^{-2}	-1.00×10^{-2}	-0.88×10^{-2}
$\hat{g}(k)$	1.80	1.28	0.29	−0.16	0.08	0.44	−0.31	0.0

2.6 小　结

本章包括辨识的理论基础和古典辨识法两部分内容，在 2.1～2.4 节辨识的理论基础部分中，主要介绍了随机过程的基本概念及其数学描述、谱密度与相关函数、线性系统在随机输入下的响应、随机序列和白噪声及其产生方法。系统辨识所用的数据通常含有噪声，如果这种噪声相关性较弱或者强度很小，则可以近似将其视为白噪声。因此，白噪声是一类非常重要的随机过程，它的基本概念及其产生方法与系统辨识密切相关。因此，这里不仅重点讨论了用乘同余法产生随机序列和产生白噪声的理论和方法，还讨论了用移位积存器产生 M 序列的理论和方法，同时，分别开发了各种方法对应的产生随机序列、白噪声和 M 序列的三种程序。在 2.5 节古典辨识法中，介绍了相关分析法辨识和相关函数的概念，讨论了相关分析频率响应和相关分析脉冲响应法辨识，并列举了相关分析脉冲响应法辨识的实例。

习　题

1. 简述互相关函数与互协方差函数的主要性质。
2. 系统辨识时，持续激励信号的特点是什么？
3. 随机噪声和白噪声的主要区别是什么？
4. 试利用乘同余法，选 $R=10, A=3, k=4, M=10^k=10000$，递推 80 次，采用 MATLAB 的仿真语言(m 软件)编程，产生(0,1)均匀分布随机数，打印出程序和运行结果。

5. 仍用乘同余法，选 $R=10, A=3, k=4, M=10^k=10000$，递推 80 次，采用 MATLAB 的仿真语言(m 软件)编程，产生($-1,1$)均匀分布白噪声序列，并比较在同样条件下，产生(0,1)均匀分布随机数和产生($-1,1$)均匀分布白噪声序列的程序主要区别有哪些?
6. 试说明什么是伪随机信号?
7. 用移位寄存器产生 M 序列$\{M(k)\}$时，M 序列有哪些特点? 怎样确定 M 序列的循环长度为N_p? 试解释$\{M(k)\}$的“游程”和“段”的含义?
8. 自相关函数和互相关函数的定义主要区别是什么?
9. 在相关分析法辨识中，激励(输入)信号用$\{M(k)\}$，利用互相关函数计算系统的脉冲响应方法，说明递推算法和一次完成算法有和区别?
10. 依据例 2.6 的条件，结合例 2.3，选用 10 级移位寄存器产生 $N_p=1023$ 的 M 序列，用该 M 序列作输入信号的递推算法，采用 MATLAB 的仿真语言(m 软件)编程，求加热炉的脉冲响应，并打印出结果。

第3章 最小二乘参数辨识

最小二乘法是一种经典的数据处理方法。本章将研究最小二乘类参数辨识方法,主要包括最小二乘参数估计的一次完成算法、最小二乘递推算法、增广最小二乘法、广义最小二乘法和多级最小二乘法。其中最小二乘的一次完成算法是最基本的,也是应用最广泛的一种方法,其余的方法都是以最小二乘原理为基础推导出来的。

3.1 最小二乘法的概念

早在1795年,著名科学家高斯就提出了最小二乘法 LSM(least squares method),并将其应用到了行星和彗星运动轨道的计算中。高斯在计算行星和彗星运动轨道时,要根据望远镜所获得的观测数据,估计描述天体运动的六个参数值。高斯认为,根据观测数据推断未知参数时,未知参数的最合适数值应是这样的数值,它使各次实际观测值和计算值之间差值的平方乘以度量其精确度的数值以后的和为最小。这就是最早的最小二乘法思想。此后,最小二乘法就被用来解决许多实际问题。针对不同用途,对最小二乘法进行修正,就出现了各种相应的最小二乘算法。

在系统辨识领域中,最小二乘法是一种基本的估计方法。最小二乘法可用于动态系统,也可用于静态系统;可用于线性系统,也可用于非线性系统;可用于离线估计,也可用于在线估计。在随机的环境下,利用最小二乘法时,并不要求观测数据提供其概率统计方面的信息,而其估计结果,却有相当好的统计特性。最小二乘法容易理解和掌握,利用最小二乘原理所拟定的辨识算法在实施上比较简单。在其他参数辨识方法难以使用时,最小二乘法能提供问题的解决方案。此外,许多用于辨识和系统参数估计的算法往往也可以解释为最小二乘法。所有这些原因使得最小二乘法广泛应用于系统辨识领域,同时最小二乘法也达到了相当完善的程度。

3.1.1 系统辨识结构

考虑随机模型的参数估计问题时,首先考虑单输入单输出系统 SISO(single input single output)。如图3.1所示,把待辨识的系统看做“黑箱”,它只考虑系统的输入输出特性,而不强调系统的内部机理。图中,输入 $u(k)$和输出 $z(k)$是可以观测的;$G(z^{-1})$是系统模型,用来描述系统的输入输出特

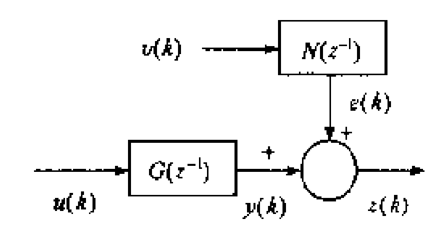


图3.1 SISO系统的“黑箱”结构

性;$N(z^{-1})$是噪声模型,$v(k)$是白噪声,$e(k)$是有色噪声,根据表示定理,可以表示成

$$e(k)=N(z^{-1})v(k) \tag{3.1}$$

通常

$$G(z^{-1})=\frac{B(z^{-1})}{A(z^{-1})},\quad N(z^{-1})=\frac{D(z^{-1})}{C(z^{-1})} \tag{3.2}$$

式中

$$\begin{cases}A(z^{-1})=1+a_1z^{-1}+a_2z^{-2}+\cdots+a_{n_a}z^{-n_a}\\ B(z^{-1})=b_1z^{-1}+b_2z^{-2}+\cdots+b_{n_b}z^{-n_b}\end{cases} \tag{3.3}$$

$$\begin{cases}C(z^{-1})=1+c_1z^{-1}+c_2z^{-2}+\cdots+c_{n_c}z^{-n_c}\\ D(z^{-1})=1+d_1z^{-1}+d_2z^{-2}+\cdots+d_{n_d}z^{-n_d}\end{cases} \tag{3.4}$$

本章所用的系统模型总是一样的,噪声模型在各个方法中有所不同。对实际辨识问题来说,应该选用什么样的模型,这没有一般的原则可循。总的来说,可先选择简单的模型获得辨识的结果,检验模型的可信度,或者看实际的使用效果。如果不能满足要求,就需要换用他种模型,这时所用的辨识方法自然也就不同。也就是说,解决一个实际问题,到底应该采用哪种辨识方法,这要取决于模型类的选择,而模型类的确定往往需要通过多次的实验比较,最后才能确认。

3.1.2 最小二乘法的基本概念

对于 SISO 离散随机系统,其描述方程为

$$\begin{aligned}&z(k)+a_1z(k-1)+\cdots+a_{n_a}z(k-n_a)\\ &=b_1u(k-1)+b_2u(k-2)\cdots+b_{n_b}u(k-n_b)+e(k)\end{aligned} \tag{3.5}$$

式中,$z(k)$为系统输出量的第 k 次观测值,$z(k-1)$为系统输出量的第 $k-1$ 次观测值,依次类推;$u(k)$为系统的第 k 个输入值,$u(k-1)$为系统的第$(k-1)$个输入值;$e(k)$是均值为零的随机噪声。将式(3.5)改写

$$\begin{aligned}z(k)=&-a_1z(k-1)-\cdots-a_{n_a}z(k-n_a)\\ &+b_1u(k-1)+b_2u(k-2)\cdots+b_{n_b}u(k-n_b)+e(k)\end{aligned} \tag{3.6}$$

可得系统输入输出的最小二乘格式

$$z(k)=\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}+e(k) \tag{3.7}$$

式中,$\boldsymbol{h}$ 为样本集合,$\boldsymbol{\theta}$ 为被辨识的参数集合。

$$\begin{cases}\boldsymbol{h}(k)=[-z(k-1),\cdots,-z(k-n_a),u(k-1),\cdots,u(k-n_b)]^{\mathrm{T}}\\ \boldsymbol{\theta}=[a_1,a_2,\cdots,a_{n_a},b_1,b_2,\cdots,b_{n_b}]^{\mathrm{T}}\end{cases} \tag{3.8}$$

取准则函数

$$J(\boldsymbol{\theta}) = \sum_{k=1}^{\infty} [e(k)]^2 = \sum_{k=1}^{\infty} [z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}]^2 \tag{3.9}$$

使 $J(\boldsymbol{\theta})=\min$ 的 $\boldsymbol{\theta}$ 估计值记作 $\hat{\boldsymbol{\theta}}_{\mathrm{LS}}$,称作参数 $\boldsymbol{\theta}$ 的最小二乘估计值。

上述基本概念表明,未知模型参数 $\boldsymbol{\theta}$ 最可能的值是在实际观测值与计算值之累次误差的平方和达到最小值处,所得到的这种模型输出能最好地接近实际系统的输出。

例3.1 考虑一个离散时间SISO系统,设作用于系统的输入序列为$\{u(1), u(2),\cdots,u(L)\}$,相应观测到的输出序列为$\{z(1),z(2),\cdots,z(L)\}$。选择下列模型

$$z(k) + az(k-1) = bu(k-1) + e(k) \tag{3.10}$$

式中,a、b 为待辨识参数。将上式写成最小二乘格式

$$z(k) = [-z(k-1),u(k-1)]\begin{bmatrix} a \\ b \end{bmatrix} + e(k) = \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta} + e(k) \tag{3.11}$$

采用准则函数

$$J(\boldsymbol{\theta}) = \sum_{k=1}^{\infty} [e(k)]^2 = \sum_{k=1}^{\infty} [z(k) + az(k-1) - bu(k-1)]^2 \tag{3.12}$$

根据输入输出数据,极小化 J,求参数 a、b 使得$J(\boldsymbol{\theta})=\min$。这就是最小二乘辨识问题。

3.2 最小二乘问题的描述

设时不变SISO系统的数学模型为

$$A(z^{-1})z(k) = B(z^{-1})u(k) + e(k) \tag{3.13}$$

式中,$u(k)$和 $z(k)$为系统输入输出量;$e(k)$是噪声;多项式 $A(k)$和 $B(k)$如式(3.3)所示。现在的问题是如何利用系统的输入输出数据,确定多项式 $A(k)$和$B(k)$的系数。

在解决这类系统辨识问题之前,先明确一些基本假设和基本关系。首先假定系统模型(3.13)的阶次 n_a 和 n_b 已经设定,且一般有 $n_a > n_b$。当取相同阶次时,记作 $n = n_a = n_b$。其次,将模型(3.13)写成最小二乘格式

$$z(k) = \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta} + e(k) \tag{3.14}$$

式中

$$\begin{cases} \boldsymbol{h}(k) = [-z(k-1),\cdots,-z(k-n_a),u(k-1),\cdots,u(k-n_b)]^{\mathrm{T}} \\ \boldsymbol{\theta} = [a_1,a_2,\cdots,a_{n_a},b_1,b_2,\cdots,b_{n_b}]^{\mathrm{T}} \end{cases} \tag{3.15}$$

对于 $k=1,2,\cdots,L$，方程(3.14)构成一个线性方程组，可以把它写成

$$\boldsymbol{z}_L(k) = \boldsymbol{H}_L(k)\boldsymbol{\theta} + \boldsymbol{e}_L(k) \tag{3.16}$$

式中

$$\boldsymbol{z}_L = \begin{bmatrix} z(1) \\ z(2) \\ \vdots \\ z(L) \end{bmatrix}, \quad \boldsymbol{e}_L = \begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(L) \end{bmatrix}$$

$$\boldsymbol{H}_L = \begin{bmatrix} -z(0) & \cdots & -z(1-n_a) & u(0) & \cdots & u(1-n_b) \\ -z(1) & \cdots & -z(2-n_a) & u(1) & \cdots & u(2-n_b) \\ \vdots & & \vdots & \vdots & & \vdots \\ -z(L-1) & \cdots & -z(L-n_a) & u(L-1) & \cdots & u(L-n_b) \end{bmatrix} \tag{3.17}$$

另外，设模型(3.14)的噪声 $e(k)$ 完全可以用一阶和二阶统计矩阵描述，即设它的均值矩阵和协方差矩阵分别为

$$E\{\boldsymbol{e}_L\} = [E\{e(1)\},E\{e(2)\},\cdots,E\{e(L)\}]^{\mathrm{T}} = \boldsymbol{0} \tag{3.18}$$

$$\mathrm{cov}\{\boldsymbol{e}_L\} = E\{\boldsymbol{e}_L\boldsymbol{e}_L{}^{\mathrm{T}}\} = \begin{bmatrix} E\{e^2(1)\} & E\{e(1)e(2)\} & \cdots & E\{e(1)e(L)\} \\ E\{e(2)e(1)\} & E\{e^2(2)\} & \cdots & E\{e(2)e(L)\} \\ \vdots & \vdots & & \vdots \\ E\{e(L)e(1)\} & E\{e(L)e(2)\} & \cdots & E\{e^2(L)\} \end{bmatrix}$$

$$\triangleq \textstyle\sum_n \tag{3.19}$$

为了评价最小二乘估计的性质，还必须进一步假设噪声 $e(k)$ 是不相关的，而且是同分布的随机变量。简单的说，必须假设 $\{e(k)\}$ 是白噪声序列，即

$$\begin{cases} E\{\boldsymbol{e}_L\} = \boldsymbol{0} \\ \mathrm{cov}\{\boldsymbol{e}_L\} = \sigma_n^2\boldsymbol{I} \end{cases} \tag{3.20}$$

式中，σ_n^2 为噪声 $e(k)$ 的方差；$\boldsymbol{I}$ 为单位矩阵。有时还要假设噪声 $e(k)$ 服从正态分布。此外，还认为噪声 $e(k)$ 和输入 $u(k)$ 是不相关的，即

$$E\{e(k)u(k-l)\} = 0, \ \forall k,l \tag{3.21}$$

最后，如何选择数据长度也是要考虑的问题。显然，联立方程组(3.16)具有 L 个方程，包含 n_a+n_b 个未知数。如果 $L<n_a+n_b$，方程的个数少于未知数个数，模型参数 $\boldsymbol{\theta}$ 不能惟一确定，这种情况一般可以不去考虑它。如果 $L=n_a+n_b$，则只有当 $\boldsymbol{e}_L=\boldsymbol{0}$ 时，$\boldsymbol{\theta}$ 才有惟一的确定解。当 $\boldsymbol{e}_L\neq\boldsymbol{0}$ 时，只有取 $L>n_a+n_b$，才有可能确定一个“最优”的模型参数 $\boldsymbol{\theta}$，而且为了保证辨识的精度，L 必须充分大。

3.3 最小二乘问题的一次完成算法

3.3.1 普通最小二乘问题的解

考虑模型

$$z(k)=\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}+e(k) \tag{3.22}$$

的辨识问题,式中,$z(k)$和$\boldsymbol{h}(k)$都是可观测数据,$\boldsymbol{\theta}$是待估计参数,取准则函数

$$J(\boldsymbol{\theta})=\sum_{k=1}^{L}[e(k)]^2=\sum_{k=1}^{L}[z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}]^2=(\boldsymbol{z}_L-\boldsymbol{H}_L\boldsymbol{\theta})^{\mathrm{T}}(\boldsymbol{z}_L-\boldsymbol{H}_L\boldsymbol{\theta}) \tag{3.23}$$

极小化$J(\boldsymbol{\theta})$,求得参数$\boldsymbol{\theta}$的估计值,将使模型的输出最好地预报系统的输出。

设$\hat{\boldsymbol{\theta}}_{\mathrm{LS}}$使得$J(\boldsymbol{\theta})\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{LS}}}=\min$,则有

$$\frac{\partial J(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{LS}}}=\frac{\partial}{\partial\boldsymbol{\theta}}(\boldsymbol{z}_L-\boldsymbol{H}_L\boldsymbol{\theta})^{\mathrm{T}}(\boldsymbol{z}_L-\boldsymbol{H}_L\boldsymbol{\theta})=0 \tag{3.24}$$

展开上式,并运用如下两个向量微分公式

$$\frac{\partial}{\partial\boldsymbol{x}}(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x})=\boldsymbol{a}^{\mathrm{T}} \tag{3.25a}$$

$$\frac{\partial}{\partial\boldsymbol{x}}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x})=2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A},\quad \boldsymbol{A}\text{ 为对称阵} \tag{3.25b}$$

得正则方程

$$(\boldsymbol{H}_L{}^{\mathrm{T}}\boldsymbol{H}_L)\hat{\boldsymbol{\theta}}_{\mathrm{LS}}=\boldsymbol{H}_L{}^{\mathrm{T}}\boldsymbol{z}_L \tag{3.26}$$

当$\boldsymbol{H}_L{}^{\mathrm{T}}\boldsymbol{H}_L$是正则矩阵时,有

$$\hat{\boldsymbol{\theta}}_{\mathrm{LS}}=(\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{H}_L)^{-1}\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{z}_L \tag{3.27}$$

且

$$\frac{\partial^2 J(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2}\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{LS}}}=2\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{H}_L>0 \tag{3.28}$$

所以满足式(3.27)的$\hat{\boldsymbol{\theta}}_{\mathrm{LS}}$使$J(\boldsymbol{\theta})\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{LS}}}=\min$,并且是惟一的。

通过极小化式(3.23)计算$\hat{\boldsymbol{\theta}}_{\mathrm{LS}}$的方法称作最小二乘法,对应的$\hat{\boldsymbol{\theta}}_{\mathrm{LS}}$称为最小二乘估计值。

3.3.2 加权最小二乘问题的解

研究加权最小二乘WLS(weighted least squares)问题时,考虑模型

$$z(k)=\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}+e(k) \tag{3.29}$$

取准则函数

$$J(\boldsymbol{\theta})=\sum_{k=1}^{L}\boldsymbol{\Lambda}(k)[e_L]^2=\sum_{k=1}^{L}\boldsymbol{\Lambda}(k)[z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}]^2$$
$$=(\boldsymbol{z}_L-\boldsymbol{H}_L\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{\Lambda}_L(\boldsymbol{z}_L-\boldsymbol{H}_L\boldsymbol{\theta}) \tag{3.30}$$

式中,$\boldsymbol{\Lambda}(k)$称为加权因子,对所有的 k,$\boldsymbol{\Lambda}(k)$都必须是正数。引入加权因子的目的是为了便于考虑观测数据的可信度。如果有理由认为现在时刻的数据比过去时刻的数据可靠,那么现在时刻的加权值就要大于过去时刻的加权值。比如可选 $\boldsymbol{\Lambda}(k)=\mu^{L-k}$,$0<\mu<1$。当 $k=1$ 时,$\boldsymbol{\Lambda}(1)=\mu^{L-1}\ll 1$;当 $k=L$ 时,$\boldsymbol{\Lambda}(L)=1$,这就体现了对不同时刻的数据给予了不同程度的信任。$\boldsymbol{\Lambda}(k)$的选择,取决于人的主观因素,并无一般规律可循。在实际应用中,如果对象是线性时不变系统,或者数据的可信度还难以确定,则可以简单地选择 $\boldsymbol{\Lambda}(k)=1,\forall k$。若在一定条件下,根据噪声的方差对 $\boldsymbol{\Lambda}(k)$进行最佳选择,得到的估计值称作 Markov 估计。

根据式(3.17)的定义,准则函数 $J(\boldsymbol{\theta})$的二次型形式为

$$J(\boldsymbol{\theta})=(\boldsymbol{z}_L-\boldsymbol{H}_L\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{\Lambda}_L(\boldsymbol{z}_L-\boldsymbol{H}_L\boldsymbol{\theta}) \tag{3.31}$$

式中,加权阵 $\boldsymbol{\Lambda}_L$ 一般是正定矩阵,它与加权因子的关系是

$$\boldsymbol{\Lambda}_L=\mathbf{diag}[\boldsymbol{\Lambda}(1),\boldsymbol{\Lambda}(2),\cdots,\boldsymbol{\Lambda}(L)] \tag{3.32}$$

显然,式(3.31)中 $\boldsymbol{H}_L\boldsymbol{\theta}$ 代表模型的输出,或者说是系统输出的预报值。$J(\boldsymbol{\theta})$可以被看做用来衡量模型输出与实际系统输出的接近情况。极小化 $J(\boldsymbol{\theta})$,求得参数 $\boldsymbol{\theta}$ 的估计值,将使模型的输出最好地预报系统的输出。

设$\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}$使得 $J(\boldsymbol{\theta})\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}}=\min$,则有

$$\frac{\partial J(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}}=\frac{\partial}{\partial\boldsymbol{\theta}}(\boldsymbol{z}_L-\boldsymbol{H}_L\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{\Lambda}_L(\boldsymbol{z}_L-\boldsymbol{H}_L\boldsymbol{\theta})=0 \tag{3.33}$$

展开上式得正则方程

$$(\boldsymbol{H}_L{}^{\mathrm{T}}\boldsymbol{\Lambda}_L\boldsymbol{H}_L)\,\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}=\boldsymbol{H}_L{}^{\mathrm{T}}\boldsymbol{\Lambda}_L\boldsymbol{z}_L \tag{3.34}$$

当 $\boldsymbol{H}_L{}^{\mathrm{T}}\boldsymbol{\Lambda}_L\boldsymbol{H}_L$ 是正则矩阵时,有

$$\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}=(\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{\Lambda}_L\boldsymbol{H}_L)^{-1}\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{\Lambda}_L\boldsymbol{z}_L \tag{3.35}$$

且

$$\frac{\partial^2 J(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2}\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}}=2\boldsymbol{H}_L{}^{\mathrm{T}}\boldsymbol{\Lambda}_L\boldsymbol{H}_L>0 \tag{3.36}$$

所以满足式(3.35)的$\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}$使 $J(\boldsymbol{\theta})\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}}=\min$,并且是惟一的。

通过极小化式(3.31),计算$\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}$的方法称作加权最小二乘法,对应的$\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}$称为加权最小二乘估计值。如果加权阵取单位阵 $\boldsymbol{\Lambda}_L=\boldsymbol{I}$,式(3.35)退化为式

(3.27)。可以看出最小二乘法是加权最小二乘法的一种特例。

当获得一批数据后,利用式(3.27)或(3.35)可一次求得相应的参数估计值,这样处理问题的方法就称作一次完成算法或"整批"算法。它在理论研究方面有许多方便之处,但当矩阵的维数增加时,矩阵求逆运算的计算量会急剧增加,这会给计算机的计算速度和存储量带来负担,因此有时也可用高斯消元法直接解正则方程(3.26)和(3.34),以便更快地求得参数的估计值。另外,一次完成算法要求$\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{\Lambda}_L\boldsymbol{H}_L$必须是正则矩阵(可逆矩阵),其充分必要条件是系统的输入信号必须是$2n$阶持续激励信号。这就意味着辨识所用的输入信号不能随意选择,否则可能造成不能辨识。目前常用的信号有:

(1) 随机序列(如白噪声);

(2) 伪随机序列(如 M 序列或逆 M 序列);

(3) 离散序列,通常指对含有 n 种频率(各频率不能满足整数倍关系)的正弦组合信号进行采样处理获得的离散序列。

例 3.2　写出系统脉冲响应的最小二乘一次完成算法估计值。

设线性系统的输出 $z(k)$用输入序列$\{u(k)\}$与脉冲响应序列$\{g(i), i=0, 1,\cdots,N\}$的卷积和形式表示

$$z(k) = \sum_{i=0}^{N} g(i)u(k-i) + w(k) \tag{3.37}$$

式中,$w(k)$是系统输出测量噪声,设它是均值为零的白噪声。

当取 $k=0,1,\cdots,L$ 时,式(3.37)可写成式(3.16)的形式

$$\boldsymbol{z}_L(k) = \boldsymbol{H}_L(k)\boldsymbol{g} + \boldsymbol{w}_L(k) \tag{3.38}$$

式中

$$\boldsymbol{z}_L = \begin{bmatrix} z(1) \\ z(2) \\ \vdots \\ z(L) \end{bmatrix},\quad \boldsymbol{g} = \begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(N) \end{bmatrix},\quad \boldsymbol{w}_L = \begin{bmatrix} w(1) \\ w(2) \\ \vdots \\ w(L) \end{bmatrix}$$

$$\boldsymbol{H}_L = \begin{bmatrix} u(1) & u(0) & \cdots & u(1-N) \\ u(2) & u(1) & \cdots & u(2-N) \\ \vdots & \vdots & & \vdots \\ u(L) & u(L-1) & \cdots & u(L-N) \end{bmatrix} \tag{3.39}$$

$\boldsymbol{g}$ 为待辨识参数,$\boldsymbol{z}_L$ 和 $\boldsymbol{H}_L$ 都是可观测数据,根据式(3.37),系统脉冲响应的最小二乘一次完成算法的估计值为

$$\hat{\boldsymbol{g}}_{\mathrm{LS}} = (\boldsymbol{H}_L{}^{\mathrm{T}}\boldsymbol{H}_L)\boldsymbol{H}_L{}^{\mathrm{T}}\boldsymbol{z}_L \tag{3.40}$$

值得注意的是,如果噪声序列是零均值的,而且输入输出数据向量和噪声是相

互独立的，则最小二乘估计或加权最小二乘估计是无偏差估计；否则最小二乘估计将不再是无偏估计。

3.4 最小二乘一次完成算法的 MATLAB 仿真

例 3.3 考虑仿真对象

$$z(k) - 1.5z(k-1) + 0.7z(k-2) = u(k-1) + 0.5u(k-2) + v(k) \tag{3.41}$$

式中，$v(k)$是服从正态分布的白噪声 $\mathcal{N}(0,1)$。输入信号采用 4 阶 M 序列，幅度为 1。选择如下形式的辨识模型

$$z(k) + a_1 z(k-1) + a_2 z(k-2) = b_1 u(k-1) + b_2 u(k-2) + v(k) \tag{3.42}$$

按式(3.17)构造 $\boldsymbol{z}_L$和$\boldsymbol{H}_L$；数据长度取 $L=14$；加权阵取 $\boldsymbol{\Lambda}_L=\boldsymbol{I}$；利用式(3.35)或(3.27)计算参数估计值$\hat{\boldsymbol{\theta}}_{\text{LS}}$。

设输入信号的取值是从 $k=1$ 到 $k=16$ 的 M 序列，则待辨识参数$\hat{\boldsymbol{\theta}}_{\text{LS}}$为$\hat{\boldsymbol{\theta}}_{\text{LS}} = (\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{H}_L)^{-1}\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{z}_L$。式中，被辨识参数$\hat{\boldsymbol{\theta}}_{\text{LS}}$、观测矩阵 $\boldsymbol{z}_L$、$\boldsymbol{H}_L$的表达式为

$$\hat{\boldsymbol{\theta}}_{\text{LS}} = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix}, \boldsymbol{z}_L = \begin{bmatrix} z(3) \\ z(4) \\ \vdots \\ z(16) \end{bmatrix}, \boldsymbol{H}_L = \begin{bmatrix} -z(2) & -z(1) & u(2) & u(1) \\ -z(3) & -z(2) & u(3) & u(2) \\ \vdots & \vdots & \vdots & \vdots \\ -z(15) & -z(14) & u(15) & u(14) \end{bmatrix} \tag{3.43}$$

程序框图如图 3.2 所示。MATLAB 6.0 仿真程序如下：

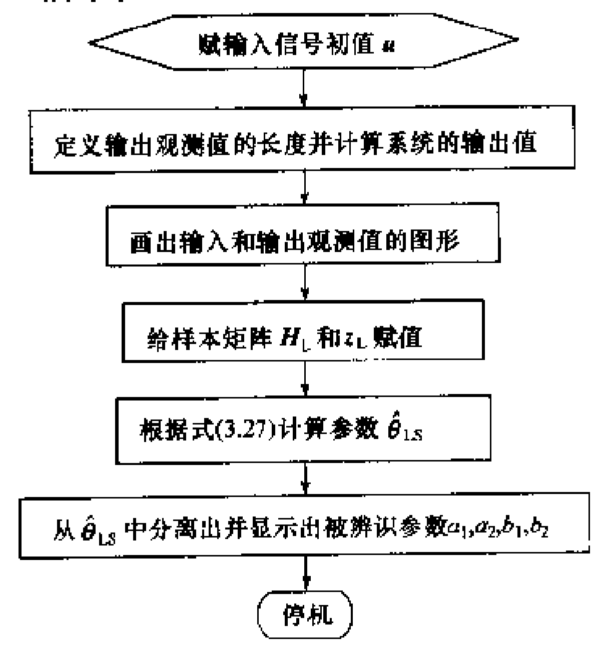


图 3.2 最小二乘一次完成算法程序框图

```
%二阶系统的最小二乘一次完成算法辨识程序,在光盘中的文件名:FLch3LSeg1.m
u=[-1,1,-1,1,1,1,1,-1,-1,-1,1,-1,-1,1,1]; %系统辨识的输入信号为一个周期的M序列
z=zeros(1,16); %定义输出观测值的长度
for k=3:16
    z(k)=1.5*z(k-1)-0.7*z(k-2)+u(k-1)+0.5*u(k-2); %用理想输出值作为观测值
end
subplot(3,1,1) %画三行一列图形窗口中的第一个图形
stem(u) %画输入信号 u 的径线图形
subplot(3,1,2) %画三行一列图形窗口中的第二个图形
i=1:1:16; %横坐标范围是1到16,步长为1
plot(i,z) %图形的横坐标是采样时刻 i, 纵坐标是输出观测值 z, 图形格式为连续曲线
subplot(3,1,3) %画三行一列图形窗口中的第三个图形
stem(z),grid on %画出输出观测值 z 的径线图形,并显示坐标网格
u,z %显示输入信号和输出观测信号
%L=14 %数据长度
HL=[-z(2) -z(1) u(2) u(1);-z(3) -z(2) u(3) u(2);-z(4) -z(3) u(4) u(3);-z(5) -z(4) u(5) u(4);
%-z(6) -z(5) u(6) u(5);-z(7) -z(6) u(7) u(6);-z(8) -z(7) u(8) u(7);-z(9) -z(8) u(9) u(8);
%-z(10) -z(9) u(10) u(9);-z(11) -z(10) u(11) u(10);-z(12) -z(11) u(12) u(11);-z(13) -z(12)
%u(13) u(12);-z(14) -z(13) u(14) u(13);-z(15) -z(14) u(15) u(14)] %给样本矩阵 H_L 赋值
ZL=[z(3);z(4);z(5);z(6);z(7);z(8);z(9);z(10);z(11);z(12);z(13);z(14);z(15);z(16)] % 给样本矩阵
z_L赋值
%Calculating Parameters

c1=HL'*HL; c2=inv(c1); c3=HL'*ZL; c=c2*c3 %计算并显示θ̂_LS
%Display Parameters

a1=c(1), a2=c(2), b1=c(3),b2=c(4) %从θ̂_LS中分离出并显示 a1、a2、b1、b2
%End
```

程序运行结果:

```
>>
u=[-1,1,-1,1,1,1,1,-1,-1,-1,1,-1,-1,1,1]
z=[0,0,0.5000,0.2500,0.5250,2.1125,4.3012,6.4731,6.1988,3.2670,-0.9386,-3.1949,-4.6352,
6.2165,-5.5800,-2.5185]
```

$$
HL=\begin{bmatrix}
0 & 0 & 1.0000 & -1.0000 \\
-0.5000 & 0 & -1.0000 & 1.0000 \\
-0.2500 & -0.5000 & 1.0000 & -1.0000 \\
-0.5250 & -0.2500 & 1.0000 & 1.0000 \\
-2.1125 & -0.5250 & 1.0000 & 1.0000 \\
-4.3012 & -2.1125 & 1.0000 & 1.0000 \\
-6.4731 & -4.3012 & -1.0000 & 1.0000 \\
-6.1988 & -6.4731 & -1.0000 & -1.0000 \\
-3.2670 & -6.1988 & -1.0000 & -1.0000 \\
0.9386 & -3.2670 & 1.0000 & -1.0000 \\
3.1949 & 0.9386 & -1.0000 & 1.0000 \\
4.6352 & 3.1949 & -1.0000 & -1.0000 \\
6.2165 & 4.6352 & 1.0000 & -1.0000 \\
5.5800 & 6.2165 & 1.0000 & 1.0000
\end{bmatrix}
$$

ZL = [0.5000, 0.2500, 0.5250, 2.1125, 4.3012, 6.4731, 6.1988, 3.2670, -0.9386, -3.1949, -4.6352, -6.2165, -5.5800, -2.5185]^T

c = [-1.5000, 0.7000, 1.0000, 0.5000]^T

a1 = -1.5000

a2 = 0.7000

b1 = 1.0000

b2 = 0.5000

程序运行曲线如图 3.3 所示。

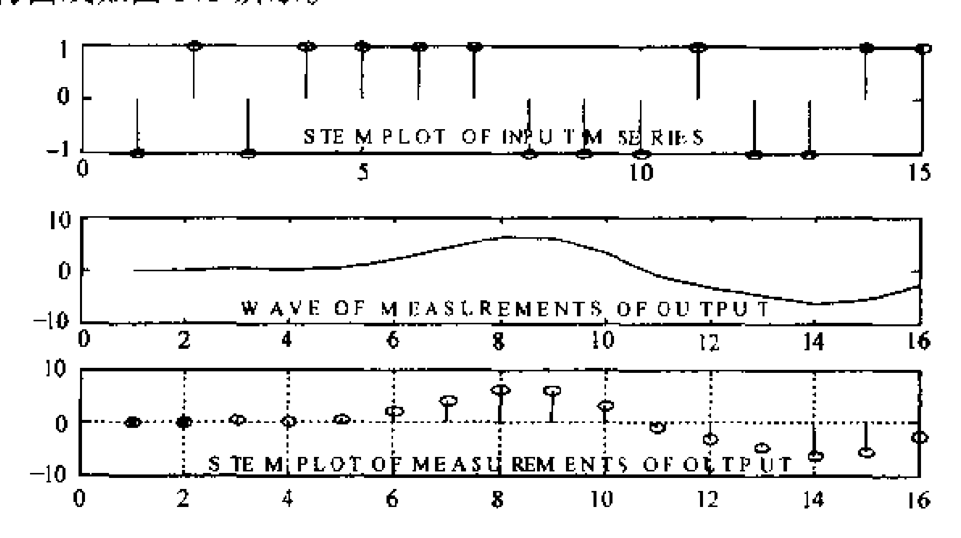


图 3.3 最小二乘一次完成算法仿真实例中输入信号和输出观测值

从仿真结果表 3.1 可以看出，由于所用的输出观测值没有任何噪声成分，所以辨识结果也无任何误差。

表 3.1 最小二乘一次完成算法的辨识结果

参数	a_1	a_2	b_1	b_2
真值	-1.5	0.7	1.0	0.5
估计值	-1.5	0.7	1.0	0.5

例 3.4 根据热力学原理，对给定质量的气体，体积 V 与压力 P 之间的关系为 $PV^{\alpha}=\beta$，其中 α 和 β 为待定参数。经实验获得如下一批数据，V 的单位为立方英寸，P 的单位为帕每平方英寸。

V 54.3 61.8 72.4 88.7 118.6 194.0

P 61.2 49.5 37.6 28.4 19.2 10.1

试用最小二乘一次完成算法确定参数 α 和 β。

首先要写出系统的最小二乘表达式。为此，把体积 V 与压力 P 之间的关系 $PV^{\alpha}=\beta$ 改为对数关系，即，$\log P=-\alpha\log V+\log\beta$。此式与式(3.14)，$z(k)=\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}+e(k)$，对比可得：$z(k)=\log P$，$\boldsymbol{h}^{\mathrm{T}}(k)=[-\log V,1]$，$\boldsymbol{\theta}=[\alpha,\log\beta]^{\mathrm{T}}$。

例 3.4 的 MATLAB 6.0 程序如下：

```
%实际压力系统的最小二乘辨识程序，在光盘中的文件名：FLch3LSeg2.m
clear %工作间清零
V=[54.3,61.8,72.4,88.7,118.6,194.0]',P=[61.2,49.5,37.6,28.4,19.2,10.1]'%赋初值并显示 V、P
%logP= -alpha*logV+log(beita)=[-logV,1][alpha,log(beita)]'=HL*sita %注释 P、V之间的关系
for i=1:6; Z(i)=log(P(i)); %循环变量的取值为从 1 到 6，系统的采样输出赋值
End %循环结束
ZL=Z' % z_L赋值
HL=[-log(V(1)),1;-log(V(2)),1;-log(V(3)),1;-log(V(4)),1;-log(V(5)),1;-log(V(6)),1]
%H_L 赋值
%Calculating Parameters
c1=HL'*HL; c2=inv(c1); c3=HL'*ZL; c4=c2*c3 %计算被辨识参数的值
%Separation of Parameters
alpha=c4(1) %α 为 c4 的第一个元素
beita=exp(c4(2)) %β 为以自然数为底的 c4 的第二个元素的指数
```

程序运行结果：

```
V=[54.3000, 61.8000, 72.4000, 88.7000, 118.6000, 194.0000]'
P=[61.2000, 49.5000, 37.6000, 28.4000, 19.2000, 10.1000]'
ZL=[4.1141, 3.9020, 3.6270, 3.3464, 2.9549, 2.3125]^T
```

$$HL=\begin{bmatrix} -3.9945 & 1.0000 \\ -4.1239 & 1.0000 \\ -4.2822 & 1.0000 \\ -4.4853 & 1.0000 \\ -4.7758 & 1.0000 \\ -5.2679 & 1.0000 \end{bmatrix}$$

$$c4=\begin{bmatrix} 1.4042 \\ 9.6786 \end{bmatrix}$$

```
alpha=1.4042
beita=1.5972e+004
>>
```

仿真结果表明，用最小二乘一次完成算法可以迅速辨识出系统参数，即 $\alpha=1.4042$，$\beta=1.5972\text{e}+004$。

3.5　最小二乘参数估计的递推算法

前面给出的最小二乘一次完成算法适合于理论分析。在具体使用时，不仅占用内存量大，还不适合在线辨识。为了减少计算量，减少数据在计算机中所占的存储量，也为了有可能实时地辨识出动态系统的特性，在用最小二乘法进行参数估计时，把它化成一种既经济又有效的参数递推估计，也叫做序贯估计。

3.5.1 递推算法的概念

所谓参数递推估计，就是当被辨识系统在运行时，每取得一次新的观测数据后，就在前次估计结果的基础上，利用新引入的观测数据对前次估计的结果，根据递推算法进行修正，从而递推地得出新的参数估计值。这样，随着新的观测数据的逐次引入，一次接着一次的进行参数估计，直到参数估计值达到满意的精确程度为止。最小二乘递推算法 RLS(recursive least squares)的基本思想可以概括成

$$\text{新的估计值}\hat{\boldsymbol{\theta}}(k)=\text{老的估计值}\hat{\boldsymbol{\theta}}(k-1)+\text{修正项} \tag{3.44}$$

3.5.2 递推算法的推导

首先将式(3.35)的最小二乘一次完成算法写成

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\mathrm{WLS}}&=(\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{\Lambda}_L\boldsymbol{H}_L)^{-1}\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{\Lambda}_L\boldsymbol{z}_L=\boldsymbol{P}(L)\boldsymbol{H}_L^{\mathrm{T}}\boldsymbol{\Lambda}_L\boldsymbol{Z}_L\\&=\Big[\sum_{i=1}^{L}\boldsymbol{\Lambda}(i)\boldsymbol{h}(i)\boldsymbol{h}^{\mathrm{T}}(i)\Big]^{-1}\Big[\sum_{i=1}^{L}\boldsymbol{\Lambda}(i)\boldsymbol{h}(i)z(i)\Big]\end{aligned} \tag{3.45}$$

定义

$$\begin{cases}\boldsymbol{P}^{-1}(k)=\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{\Lambda}_k\boldsymbol{H}_k=\sum\limits_{i=1}^{k}\Lambda(i)\boldsymbol{h}(i)\boldsymbol{h}^{\mathrm{T}}(i)\\ \boldsymbol{P}^{-1}(k-1)=\boldsymbol{H}_{k-1}^{\mathrm{T}}\boldsymbol{\Lambda}_{k-1}\boldsymbol{H}_{k-1}=\sum\limits_{i=1}^{k-1}\Lambda(i)\boldsymbol{h}(i)\boldsymbol{h}^{\mathrm{T}}(i)\end{cases} \tag{3.46}$$

式中，

$$\boldsymbol{H}_k=\begin{bmatrix}\boldsymbol{h}^{\mathrm{T}}(1)\\ \boldsymbol{h}^{\mathrm{T}}(2)\\ \vdots\\ \boldsymbol{h}^{\mathrm{T}}(k)\end{bmatrix},\quad \boldsymbol{\Lambda}_k=\begin{bmatrix}\boldsymbol{\Lambda}(1) & & & \boldsymbol{0}\\ & \boldsymbol{\Lambda}(2) & & \\ & & \ddots & \\ \boldsymbol{0} & & & \boldsymbol{\Lambda}(k)\end{bmatrix} \tag{3.47}$$

$$\boldsymbol{H}_{k-1}=\begin{bmatrix}\boldsymbol{h}^{\mathrm{T}}(1)\\ \boldsymbol{h}^{\mathrm{T}}(2)\\ \vdots\\ \boldsymbol{h}^{\mathrm{T}}(k-1)\end{bmatrix},\quad \boldsymbol{\Lambda}_{k-1}=\begin{bmatrix}\boldsymbol{\Lambda}(1) & & & \boldsymbol{0}\\ & \boldsymbol{\Lambda}(2) & & \\ & & \ddots & \\ \boldsymbol{0} & & & \boldsymbol{\Lambda}(k-1)\end{bmatrix} \tag{3.48}$$

$\boldsymbol{h}(i)$是一个列向量，也就是$\boldsymbol{H}_L$第i行向量的转置；$\boldsymbol{P}(k)$是一个方阵，它的维数取决于未知参数的个数，而与观测次数无关. 如果未知参数的个数是n，则$\boldsymbol{P}(k)$的维数为$n\times n$。

由式(3.46)可得

$$\boldsymbol{P}^{-1}(k)=\sum_{i=1}^{k-1}\Lambda(i)\boldsymbol{h}(i)\boldsymbol{h}^{\mathrm{T}}(i)+\Lambda(k)\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)$$

$$= \boldsymbol{P}^{-1}(k-1) + \Lambda(k)\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k) \tag{3.49}$$

设

$$\boldsymbol{z}_{k-1} = [z(1), z(2), \cdots, z(k-1)]^{\mathrm{T}} \tag{3.50}$$

则

$$\begin{aligned}\hat{\boldsymbol{\theta}}(k-1) &= (\boldsymbol{H}_{k-1}^{\mathrm{T}}\boldsymbol{\Lambda}_{k-1}\boldsymbol{H}_{k-1})^{-1}\boldsymbol{H}_{k-1}^{\mathrm{T}}\boldsymbol{\Lambda}_{k-1}\boldsymbol{z}_{k-1} \\ &= \boldsymbol{P}(k-1)\left[\sum_{i=1}^{k-1}\Lambda(i)\boldsymbol{h}(i)z(i)\right]\end{aligned} \tag{3.51}$$

于是有

$$\boldsymbol{P}^{-1}(k-1)\,\hat{\boldsymbol{\theta}}(k-1) = \sum_{i=1}^{k-1}\Lambda(i)\boldsymbol{h}(i)z(i) \tag{3.52}$$

令

$$\boldsymbol{z}_{k} = [z(1), z(2), \cdots, z(k)]^{\mathrm{T}} \tag{3.53}$$

利用式(3.49)和(3.52),可得

$$\begin{aligned}\hat{\boldsymbol{\theta}}(k) &= (\boldsymbol{H}_{k}^{\mathrm{T}}\boldsymbol{\Lambda}_{k}\boldsymbol{H}_{k})^{-1}\boldsymbol{H}_{k}^{\mathrm{T}}\boldsymbol{\Lambda}_{k}\boldsymbol{z}_{k} = \boldsymbol{P}(k)\left[\sum_{i=1}^{k}\Lambda(i)\boldsymbol{h}(i)z(i)\right] \\ &= \boldsymbol{P}(k)[\boldsymbol{P}^{-1}(k-1)\,\hat{\boldsymbol{\theta}}(k-1) + \Lambda(k)\boldsymbol{h}(k)z(k)] \\ &= \boldsymbol{P}(k)\{[\boldsymbol{P}^{-1}(k) - \boldsymbol{\Lambda}(k)\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)]\,\hat{\boldsymbol{\theta}}(k-1) + \Lambda(k)\boldsymbol{h}(k)z(k)\} \\ &= \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{P}(k)\boldsymbol{h}(k)\boldsymbol{\Lambda}(k)[z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\,\hat{\boldsymbol{\theta}}(k-1)]\end{aligned} \tag{3.54}$$

引进增益矩阵 $\boldsymbol{K}(k)$, 定义

$$\boldsymbol{K}(k) = \boldsymbol{P}(k)\boldsymbol{h}(k)\boldsymbol{\Lambda}(k) \tag{3.55}$$

则式(3.54)写成

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{K}(k)[z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\,\hat{\boldsymbol{\theta}}(k-1)] \tag{3.56}$$

进一步把式(3.49)写成

$$\boldsymbol{P}(k) = [\boldsymbol{P}^{-1}(k-1) + \Lambda(k)\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)]^{-1} \tag{3.57}$$

利用矩阵反演公式

$$(\boldsymbol{A} + \boldsymbol{C}\boldsymbol{C}^{\mathrm{T}})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{C}(\boldsymbol{I} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{C})^{-1}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{A}^{-1} \tag{3.58}$$

将式(3.57)演变成

$$\begin{aligned}\boldsymbol{P}(k) &= \boldsymbol{P}(k-1) - \boldsymbol{P}(k-1)\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\left[\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}(k) + \frac{1}{\boldsymbol{\Lambda}(k)}\right]^{-1} \\ &= \left[\boldsymbol{I} - \frac{\boldsymbol{P}(k-1)\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)}{\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}(k) + \boldsymbol{\Lambda}^{-1}(k)}\right]\boldsymbol{P}(k-1)\end{aligned} \tag{3.59}$$

将上式代入式(3.55),整理后得

$$\boldsymbol{K}(k) = \boldsymbol{P}(k-1)\boldsymbol{h}(k)\left[\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}(k) + \frac{1}{\boldsymbol{\Lambda}(k)}\right]^{-1} \tag{3.60}$$

综合式(3.56)、(3.59)和(3.60)得到加权最小二乘参数估计递推算法 RWLS(recursive weighted least squares)

$$\begin{cases}\hat{\boldsymbol{\theta}}(k)=\hat{\boldsymbol{\theta}}(k-1)+\boldsymbol{K}(k)[z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1)]\\ \boldsymbol{K}(k)=\boldsymbol{P}(k-1)\boldsymbol{h}(k)\left[\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}(k)+\dfrac{1}{\Lambda(k)}\right]^{-1}\\ \boldsymbol{P}(k)=[\boldsymbol{I}-\boldsymbol{K}(k)\boldsymbol{h}^{\mathrm{T}}(k)]\boldsymbol{P}(k-1)\end{cases}\tag{3.61}$$

式中,当 $\Lambda(k)=1,\forall k$ 时,加权最小二乘参数估计递推算法就简化成最小二乘参数估计递推算法 RLS。加权参数$\frac{1}{\Lambda}$可以在(0,1]范围内选择。如果$\frac{1}{\Lambda}=1$,意味着所有采样数据都是等同加权的,如果$\frac{1}{\Lambda}\ll 1$,则表示对新近获得的数据给予充分大的权因子,从而削弱过去的观测数据的作用。

式(3.61)表明,k 时刻的参数估计值 $\hat{\boldsymbol{\theta}}(k)$等于$(k-1)$时刻的参数估计值 $\hat{\boldsymbol{\theta}}(k-1)$加上修正项,修正项正比于 k 时刻的新息 $\tilde{z}(k)=z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1)$,其增益矩阵 $\boldsymbol{K}(k)$是时变矩阵,$\boldsymbol{P}(k)$是对称矩阵,为了保证 $\boldsymbol{P}(k)$对称性,有时把式(3.61)第三算式写成

$$\begin{aligned}\boldsymbol{P}(k)&=\boldsymbol{P}(k-1)-\frac{[\boldsymbol{P}(k-1)\boldsymbol{h}(k)][\boldsymbol{P}(k-1)\boldsymbol{h}(k)]^{\mathrm{T}}}{\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}(k)+\dfrac{1}{\Lambda(k)}}\\&=\boldsymbol{P}(k-1)-\boldsymbol{K}(k)\boldsymbol{K}^{\mathrm{T}}(k)\left[\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}(k)+\frac{1}{\Lambda(k)}\right]\end{aligned}\tag{3.62}$$

这样,在计算过程中即使有舍入误差,也能保持 $\boldsymbol{P}(k)$矩阵始终是对称的。

在最小二乘参数估计的递推算法(3.61)或(3.62)中,根据前次观测数据得到的 $\boldsymbol{P}(k-1)$及新的观测数据,可以计算出 $\boldsymbol{K}(k)$,从而由$\hat{\boldsymbol{\theta}}(k-1)$递推算出$\hat{\boldsymbol{\theta}}(k)$,下一次的递推计算所需的 $\boldsymbol{P}(k)$也可根据 $\boldsymbol{P}(k-1)$和 $\boldsymbol{K}(k)$等计算出来。在每次的递推计算过程中,信息的变换情况如图 3.4 所示。

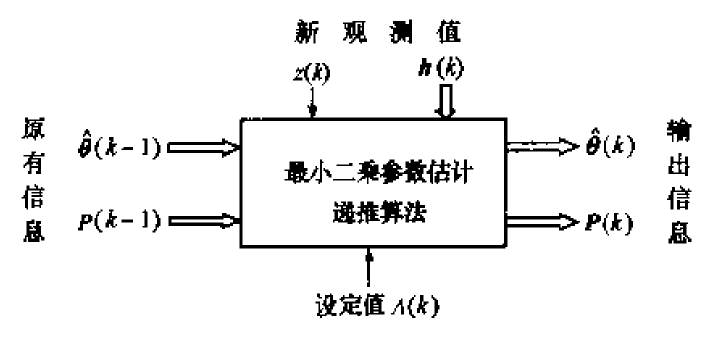


图 3.4 参数递推估计过程中信息的变换

图3.4表明,递推计算需要事先选择初始参数$\hat{\boldsymbol{\theta}}(0)$和$\boldsymbol{P}(0)$,它们的取值有两种选择方式。一是根据一批数据利用一次完成算法,预先求得

$$\begin{cases}\boldsymbol{P}(L_0) = (\boldsymbol{H}_{L_0}^{\mathrm{T}}\boldsymbol{\Lambda}_{L_0}\boldsymbol{H}_{L_0})^{-1} \\ \hat{\boldsymbol{\theta}}(L_0) = \boldsymbol{P}(0)\boldsymbol{H}_{L_0}^{\mathrm{T}}\boldsymbol{\Lambda}_{L_0}\boldsymbol{z}_{L_0}\end{cases} \tag{3.63}$$

置$\boldsymbol{P}(0)=\boldsymbol{P}(L_0)$,$\hat{\boldsymbol{\theta}}(0)=\hat{\boldsymbol{\theta}}(L_0)$,式中,$L_0$为数据长度,为了减少计算量,$L_0$不宜取太大;另一种是直接取

$$\begin{cases}\boldsymbol{P}(0) = \alpha^2\boldsymbol{I}, & \alpha\text{ 为充分大的实数} \\ \hat{\boldsymbol{\theta}}(L_0) = \boldsymbol{\varepsilon}, & \boldsymbol{\varepsilon}\text{ 为充分小的实向量}\end{cases} \tag{3.64}$$

因为

$$\begin{cases}\boldsymbol{P}^{-1}(k) = \sum_{i=1}^{k}\Lambda(i)\boldsymbol{h}(i)\boldsymbol{h}^{\mathrm{T}}(i) \\ \boldsymbol{P}^{-1}(k)\,\hat{\boldsymbol{\theta}}(k) = \sum_{i=1}^{k}\Lambda(i)\boldsymbol{h}(i)z(i)\end{cases} \tag{3.65}$$

根据式(3.45),则有

$$\begin{aligned}\hat{\boldsymbol{\theta}}(k) &= \left[\sum_{i=1}^{k}\boldsymbol{\Lambda}(i)\boldsymbol{h}(i)\boldsymbol{h}^{\mathrm{T}}(i)\right]^{-1}\left[\sum_{i=1}^{k}\boldsymbol{\Lambda}(i)\boldsymbol{h}(i)z(i)\right] \\ &= \left[\boldsymbol{P}^{-1}(0) + \sum_{i=1}^{k}\boldsymbol{\Lambda}(i)\boldsymbol{h}(i)\boldsymbol{h}^{\mathrm{T}}(i)\right]^{-1}\times\left[\boldsymbol{P}^{-1}(0)\,\hat{\boldsymbol{\theta}}(0) + \sum_{i=1}^{k}\boldsymbol{\Lambda}(i)\boldsymbol{h}(i)z(i)\right]\end{aligned} \tag{3.66}$$

另外,可用下式作为递推算法的停机标准

$$\max_{\forall i}\left|\frac{\hat{\boldsymbol{\theta}}_i(k) - \hat{\boldsymbol{\theta}}_i(k-1)}{\hat{\boldsymbol{\theta}}_i(k-1)}\right| < \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\text{ 是适当小的数} \tag{3.67}$$

它意味着当所有的参数估计值变化不大时,即可停机。最小二乘递推计算流程图如图3.5所示。

3.6 最小二乘递推算法的MATLAB仿真

例3.5 考虑图3.6所示的仿真对象,图中,$v(k)$是服从$\mathcal{N}(0,1)$分布的不相关随机噪声。且

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})}, \quad N(z^{-1}) = \frac{D(z^{-1})}{C(z^{-1})}$$

$$\begin{cases} A(z^{-1}) = 1 - 1.5a_1z^{-1} + 0.7z^{-2} = C(z^{-1}) \\ B(z^{-1}) = 1.0z^{-1} + 0.5z^{-2} \\ D(z^{-1}) = 1 \end{cases}$$

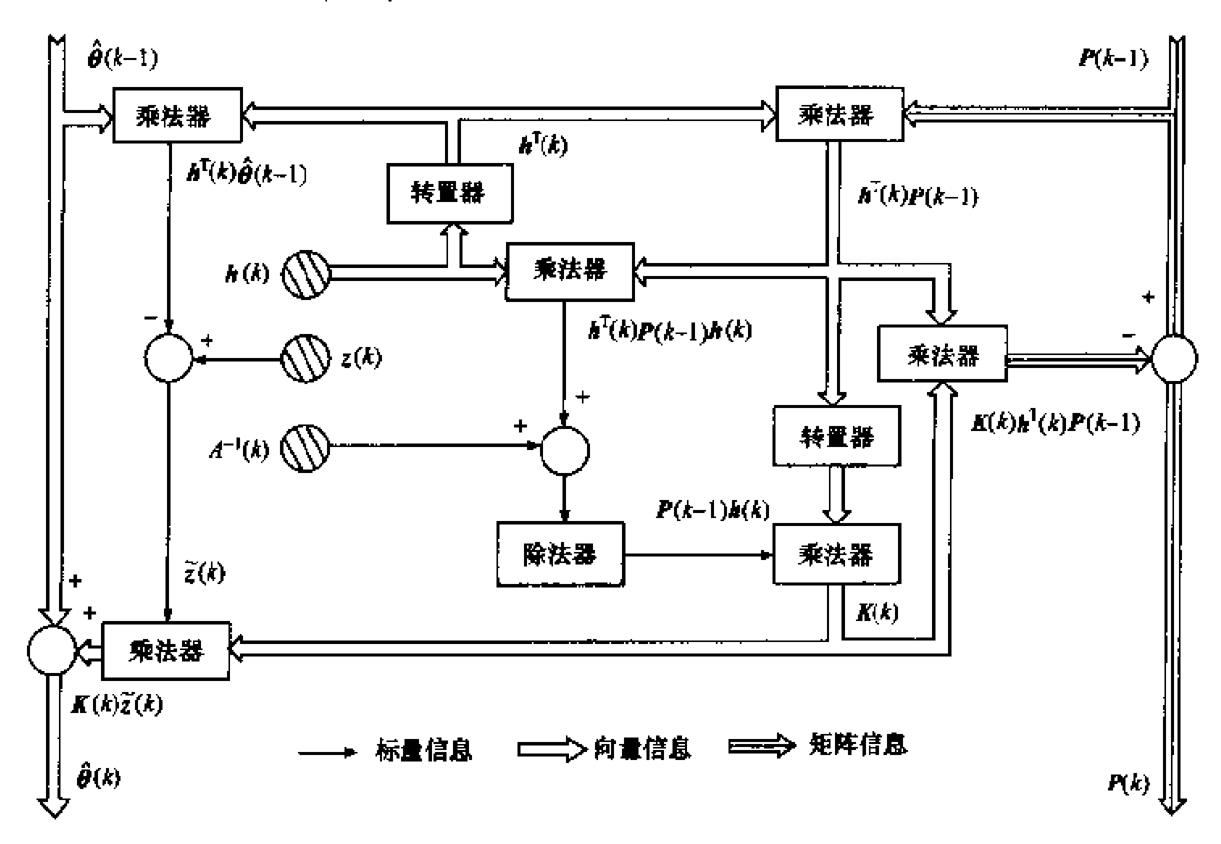


图 3.5　最小二乘递推计算流程图

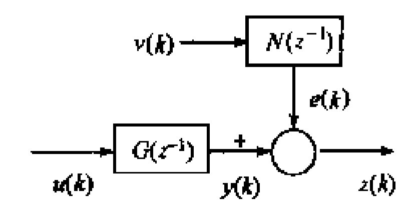


图 3.6　最小二乘递推算法辨识实例结构图

选择图 3.6 所示的辨识模型。仿真对象选择如下的模型结构

$$z(k) + a_1z(k-1) + a_2z(k-2) = b_1u(k-1) + b_2u(k-2) + v(k) \tag{3.68}$$

式中，$v(k)$是服从正态分布的白噪声 $\mathcal{N}(0,1)$。输入信号采用 4 位移位寄存器产生的 M 序列，幅度为 0.03。按式

$$z(k) - 1.5z(k-1) + 0.7z(k-2) = u(k-1) + 0.5u(k-2) + v(k) \tag{3.69}$$

构造 $\boldsymbol{h}(k)$；加权阵取单位阵 $\boldsymbol{\Lambda}_L = \boldsymbol{I}$；利用式(3.61)计算 $\boldsymbol{K}(k)$、$\hat{\boldsymbol{\theta}}(k)$和 $\boldsymbol{P}(k)$，计

算各次参数辨识的相对误差,精度满足要求式(3.67)后停机。

最小二乘递推算法辨识的 MATLAB 6.0 程序流程如图 3.7 所示。下面给出具体程序。

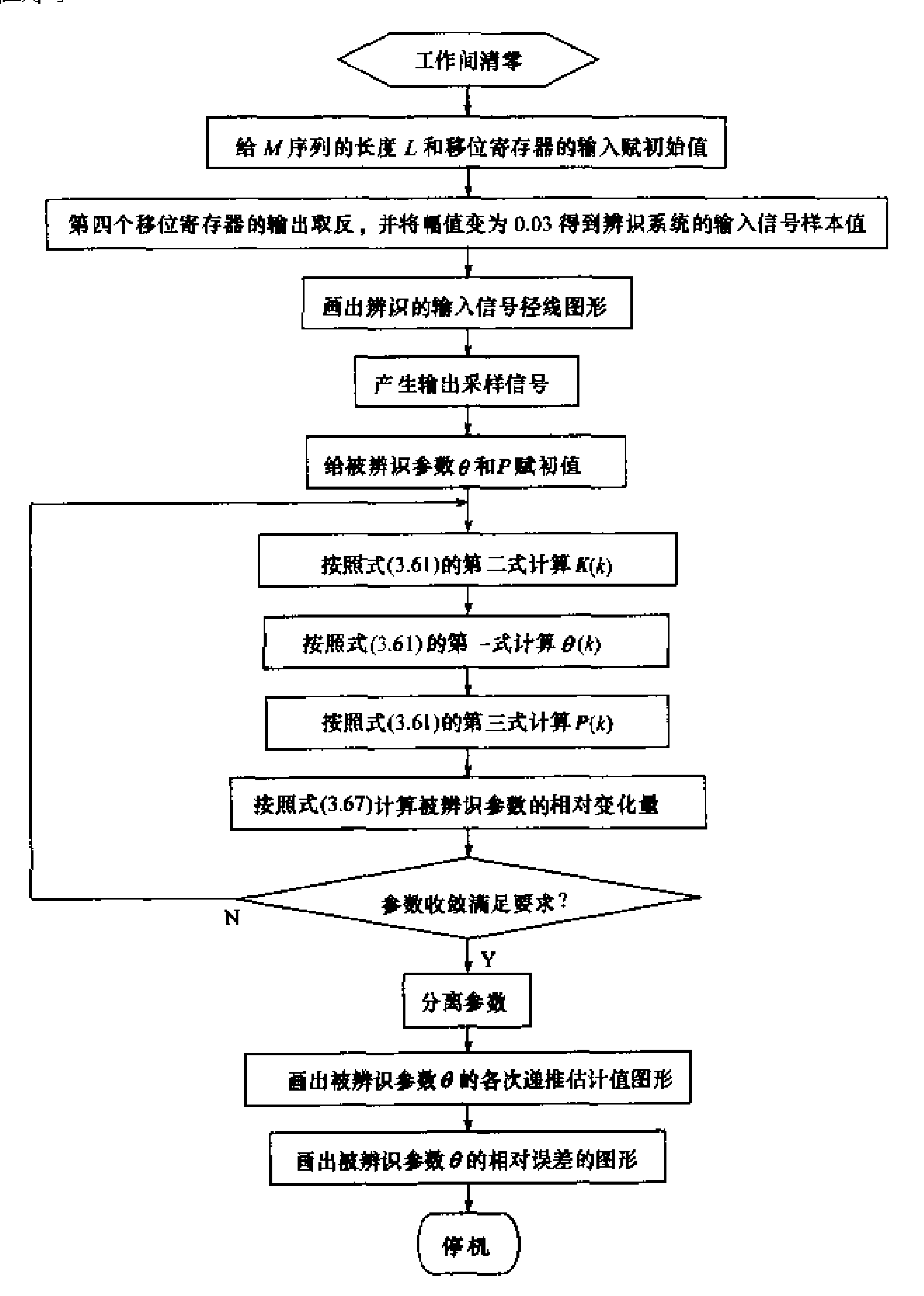


图 3.7　最小二乘递推算法辨识的 MATLAB 6.0 程序流程图

```
%最小二乘递推算法辨识程序，在光盘中的文件名:FLch3RLSeg3.m
%FLch3RLSeg3
clear %清理工作间变量
L=15; % M序列的周期
y1=1;y2=1;y3=1;y4=0; %四个移位寄存器的输出初始值
for i=1:L;%开始循环,长度为L
    x1=xor(y3,y4); %第一个移位寄存器的输入是第三个与第四个移位寄存器的输出的"或"
    x2=y1; %第二个移位寄存器的输入是第一个移位寄存器的输出
    x3=y2; %第三个移位寄存器的输入是第二个移位寄存器的输出
    x4=y3; %第四个移位寄存器的输入是第三个移位寄存器的输出
    y(i)=y4; %取出第四个移位寄存器的幅值为"0"和"1"的输出信号,即M序列
    if y(i)>0.5,u(i)=-0.03; %如果M序列的值为"1",辨识的输入信号取"-0.03"
    else u(i)=0.03; %如果M序列的值为"0",辨识的输入信号取"0.03"
    end %小循环结束
    y1=x1;y2=x2;y3=x3;y4=x4; %为下一次的输入信号做准备
end %大循环结束.产生输入信号u
figure(1); %第一个图形
stem(u),grid on %显示出输入信号径线图并给图形加上网格
z(2)=0;z(1)=0; %设z的前两个初始值为零
for k=3:15; %循环变量从3到15
    z(k)=1.5*z(k-1)-0.7*z(k-2)+u(k-1)+0.5*u(k-2); %输出采样信号
end
%RLS递推最小二乘辨识
c0=[0.001 0.001 0.001 0.001]'; %直接给出被辨识参数的初始值,即一个充分小的实向量
p0=10^6*eye(4,4); %直接给出初始状态P0,即一个充分大的实数单位矩阵
E=0.000000005; %取相对误差E=0.000000005
c=[c0,zeros(4,14)]; %被辨识参数矩阵的初始值及大小
e=zeros(4,15); %相对误差的初始值及大小
for k=3:15; %开始求K
    h1=[-z(k-1),-z(k-2),u(k-1),u(k-2)]'; x=h1'*p0*h1+1; x1=inv(x); %开始求K(k)
    k1=p0*h1*x1;%求出K的值
    d1=z(k)-h1'*c0; c1=c0+k1*d1; %求被辨识参数c
    e1=c1-c0; %求参数当前值与上一次的值的差值
    e2=e1./c0; %求参数的相对变化
    e(:,k)=e2; %把当前相对变化的列向量加入误差矩阵的最后一列
    c0=c1; %新获得的参数作为下一次递推的旧参数
    c(:,k)=c1; %把辨识参数c列向量加入辨识参数矩阵的最后一列
    p1=p0-k1*k1'*[h1'*p0*h1+1]; %求出p(k)的值
    p0=p1; %给下次用
    if e2<=E break; %如果参数收敛情况满足要求,终止计算
    end %小循环结束
```

```
end %大循环结束
c,e %显示被辨识参数及其误差(收敛)情况
%分离参数
a1=c(1,:); a2=c(2,:); b1=c(3,:); b2=c(4,:); ea1=e(1,:); ea2=e(2,:); eb1=e(3,:); eb2=e(4,:);
figure(2); %第二个图形
i=1:15; %横坐标从 1 到 15
plot(i,a1,'r',i,a2,':',i,b1,'g',i,b2,':') %画出 a1,a2,b1,b2 的各次辨识结果
title('Parameter Identification with Recursive Least Squares Method') %图形标题
figure(3); %第三个图形
i=1:15; %横坐标从 1 到 15
plot(i,ea1,'r',i,ea2,'g',i,eb1,'b',i,eb2,'r:') %画出 a1,a2,b1,b2 的各次辨识结果的收敛情况
title('Identification Precision') %图形标题
```

程序运行结果：

```
>>
c=
0.0010  0   0.0010  -0.4984  -1.2328  -1.4951  -1.4962  -1.4991
0.0010  0   0.0010   0.0010  -0.2350   0.6913   0.6941   0.6990
0.0010  0   0.2509   1.2497   1.0665   1.0017   1.0020   1.0002
0.0010  0  -0.2489   0.7500   0.5668   0.5020   0.5016   0.5008

-1.4998  -1.4999  -1.5000  -1.5000  -1.5000  -1.4999  -1.4999
 0.6998   0.6999   0.7000   0.7000   0.7000   0.7000  -0.7000
 0.9999   0.9998   0.9999   0.9999   0.9999   0.9999   0.9999
 0.5002   0.5002   0.5000   0.5000   0.5000   0.5000   0.5000

e=
0  0          0  -499.4200     1.4734   0.2128   0.0007   0.0020
0  0          0          0  -235.9916  -3.9416   0.0042   0.0070
0  0   249.8612     3.9816    -0.1466  -0.0607   0.0003  -0.0018
0  0  -249.8612    -4.0136    -0.2443  -0.1143  -0.0007  -0.0016

 0.0004   0.0000   0.0001   0.0000  -0.0000  -0.0000   0.0000
 0.0012   0.0001   0.0001  -0.0000   0.0000   0.0000   0.0000
-0.0003  -0.0001   0.0001   0.0000   0.0000   0.0000  -0.0000
-0.0012  -0.0001  -0.0004   0.0000  -0.0000   0.0000  -0.0000
```

表 3.2　最小二乘递推算法的辨识结果

参　数	a_1	a_2	b_1	b_2
真　值	-1.5	0.7	1.0	0.5
估计值	-1.4999	0.7	0.9999	0.5000

最小二乘递推算法的辨识结果如表 3.2 所示，程序运行曲线如图 3.8 所示。仿真结果表明，大约递推到第十步时，参数辨识的结果基本达到稳定状态，即 $a_1=$

$-1.4999, a_2=0.7000,\ b_1=0.9999,\ b_2=0.5000$。此时,参数的相对变化量 $E \leqslant 0.000\,000\,005$。从整个辨识过程来看,精度的要求直接影响辨识的速度。虽然最终的精度可以达到很小,但开始阶段的相对误差会很大,从图 3.8(3)图形中可看出,参数的最大相对误差会达到 3 位数。

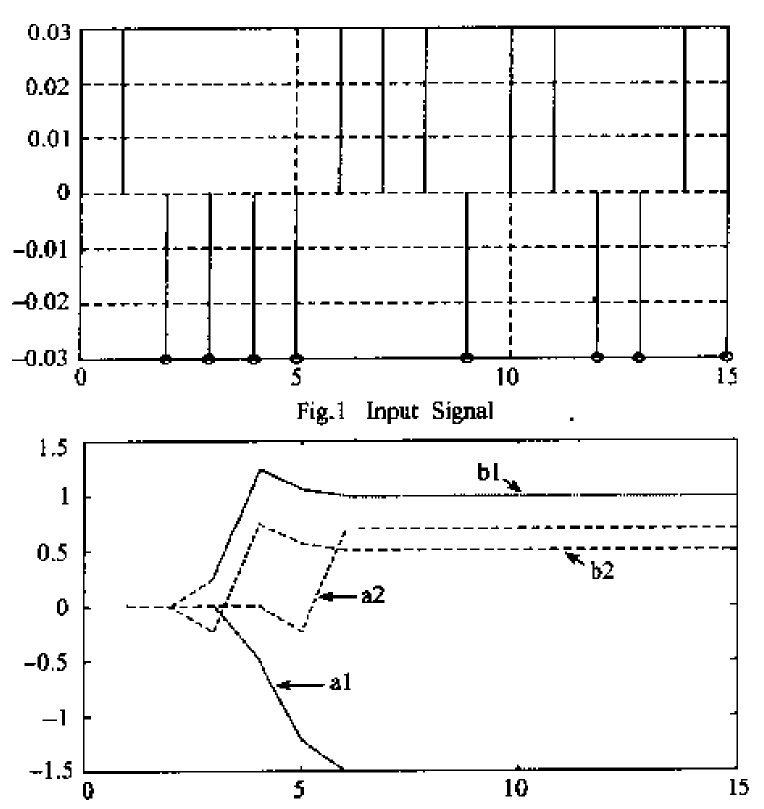


Fig.2 Parameter Iden tific ation with Recursive Least Squares Method

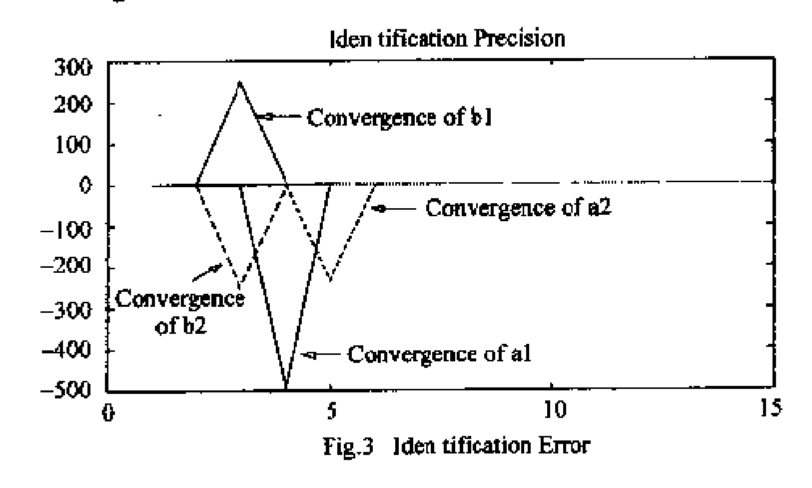


Fig.3 Iden tification Error

图 3.8 最小二乘递推算法的参数辨识仿真

3.7　增广最小二乘法

考虑 SISO 的动态系统，辨识系统的结构如图3.9所示。

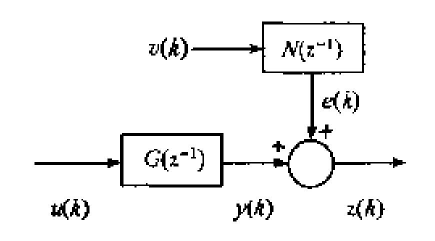


图3.9　SISO辨识系统的结构

图3.9中，输入 $u(k)$ 和输出 $z(k)$ 是可以观测的；$G(z^{-1})$ 是系统模型，用来描述系统的输入输出特性，$y(k)$ 是系统的实际输出。$N(z^{-1})$ 是噪声模型，$v(k)$ 是均值为零的不相关随机噪声。通常

$$G(z^{-1})=\frac{B(z^{-1})}{A(z^{-1})},\quad N(z^{-1})=\frac{D(z^{-1})}{C(z^{-1})} \tag{3.70}$$

式中

$$\begin{cases}A(z^{-1})=1+a_1z^{-1}+a_2z^{-2}+\cdots+a_{n_a}z^{-n_a}\\B(z^{-1})=b_1z^{-1}+b_2z^{-2}+\cdots+b_{n_b}z^{-n_b}\end{cases} \tag{3.71}$$

$$\begin{cases}C(z^{-1})=1+c_1z^{-1}+c_2z^{-2}+\cdots+c_{n_c}z^{-n_c}\\D(z^{-1})=1+d_1z^{-1}+d_2z^{-2}+\cdots+d_{n_d}z^{-n_d}\end{cases} \tag{3.72}$$

若 SISO 系统采用平均滑动模型，即

$$C(z^{-1})=A(z^{-1})$$

$$A(z^{-1})z(k)=B(z^{-1})u(k)+D(z^{-1})v(k) \tag{3.73}$$

若假定模型阶次 n_a、n_b 和 n_d 已经确定，则这类问题的辨识可用增广最小二乘法，以便获得满意的结果。令

$$\begin{cases}\boldsymbol{\theta}=[a_1,a_2,\cdots a_{n_a},b_1,b_2,\cdots b_{n_b},d_1,d_2,\cdots d_{n_d}]^{\mathrm{T}}\\\boldsymbol{h}(k)=[-z(k-1),\cdots,-z(k-n_a),\\\qquad u(k-1),\cdots,u(k-n_b),v(k-1),\cdots,v(k-n_d)]^{\mathrm{T}}\end{cases} \tag{3.74}$$

将模型(3.73)化为最小二乘格式

$$z(k)=\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}+v(k) \tag{3.75}$$

由于 $v(k)$ 是白噪声，所以利用最小二乘法即可获得参数 $\boldsymbol{\theta}$ 的无偏估计。但是数据向量 $\boldsymbol{h}(k)$ 中包含着不可测的噪声量 $v(k-1),\cdots,v(k-n_d)$，它可用相应的估计值替代。置

$$\begin{aligned}\boldsymbol{h}(k)=[&-z(k-1),\cdots,-z(k-n_a),u(k-1),\cdots,\\&u(k-n_b),v(k-1),\cdots,v(k-n_d)]^{\mathrm{T}}\end{aligned} \tag{3.76}$$

式中，$\hat{v}(k)=0,k\leqslant0$；当 $k>0$ 时，

$$\hat{v}(k)=z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1) \tag{3.77}$$

或

$$\hat{v}(k) = z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k) \tag{3.78}$$

根据式(3.61)立即就可以写出增广最小二乘递推算法 RELS(recursive extended least squares)

$$\begin{cases}\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{K}(k)[z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1)] \\ \boldsymbol{K}(k) = \boldsymbol{P}(k-1)\boldsymbol{h}(k)[\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}(k) + 1/\Lambda(k)]^{-1} \\ \boldsymbol{P}(k) = [\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{h}^{\mathrm{T}}(k)]\boldsymbol{P}(k-1)\end{cases} \tag{3.79}$$

如果$\dfrac{1}{\Lambda}=1$,即所有采样数据都是等同加权时,增广最小二乘递推算法 RELS 可以写为

$$\begin{cases}\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{K}(k)[z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1)] \\ \boldsymbol{K}(k) = \boldsymbol{P}(k-1)\boldsymbol{h}(k)[\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}(k) + 1]^{-1} \\ \boldsymbol{P}(k) = [\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{h}^{\mathrm{T}}(k)]\boldsymbol{P}(k-1)\end{cases} \tag{3.80}$$

增广最小二乘递推算法(3.79),扩充了最小二乘法的参数向量 $\boldsymbol{\theta}$ 和数据向量 $\boldsymbol{h}(k)$的维数, 把噪声模型的辨识同时考虑进去,因此被称为增广最小二乘法。最小二乘法的许多结论对它都是适用的, 但最小二乘法只能获得模型的参数估计。如果噪声模型必须用 $D(z^{-1})v(k)$表示时,只能用 RELS 算法,方可获得无偏估计,这是 RLS 算法所不能代替的。此外,以后的章节还进一步表明增广最小二乘法又是一种近似的极大似然法。

3.8 增广最小二乘辨识的 MATLAB 仿真

例 3.6 考虑图 3.10 所示的仿真对象,图中,$v(k)$是服从 $\mathcal{N}(0,1)$分布的不相关随机噪声。

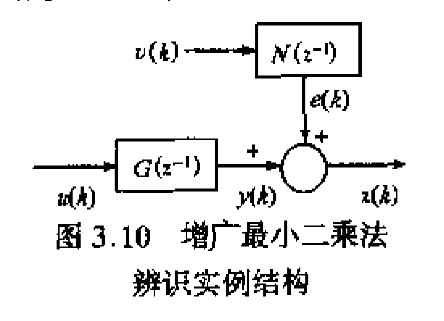


图 3.10 增广最小二乘法辨识实例结构

且

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})}, \quad N(z^{-1}) = \frac{D(z^{-1})}{C(z^{-1})}$$

$$\begin{cases}A(z^{-1}) = 1 - 1.5a_1 z^{-1} + 0.7z^{-2} = C(z^{-1}) \\ B(z^{-1}) = 1.0z^{-1} + 0.5z^{-2} \\ D(z^{-1}) = 1 - z^{-1} + 0.2z^{-2}\end{cases}$$

模型结构选用如下形式

$$\begin{aligned}&z(k) + a_1 z(k-1) + a_2(k-2) \\ &= b_1 u(k-1) + b_2 u(k-2) + v(k) + d_1 v(k-1) + d_2 v(k-2)\end{aligned}$$

增广最小二乘辨识程序流程如图 3.11 所示，递推算法的辨识结果如表 3.3 所示，程序运行曲线如图 3.12 所示，MATLAB 6.0 程序如下。

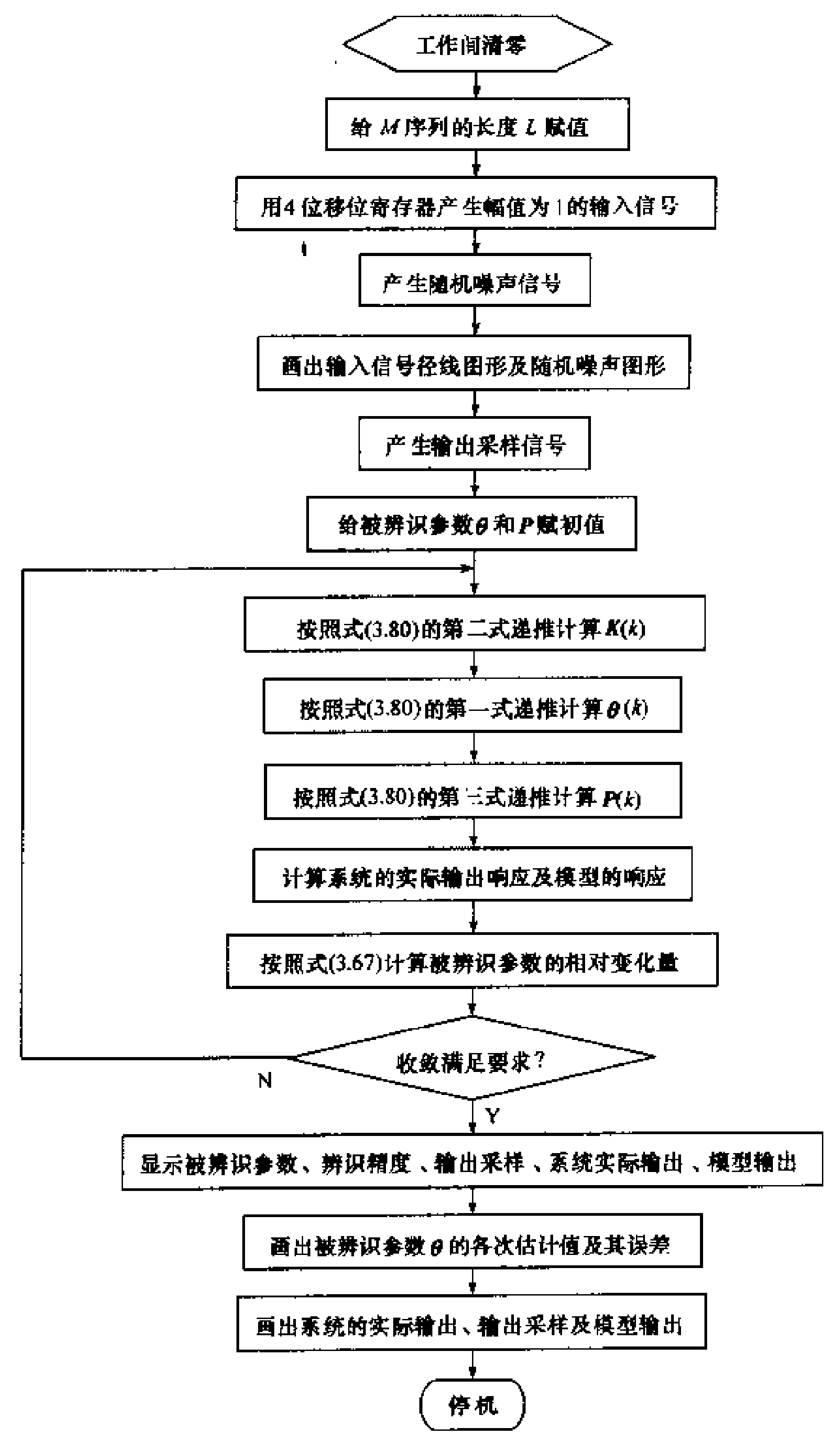


图 3.11　增广最小二乘递推算法辨识的 MATLAB 6.0 程序流程图

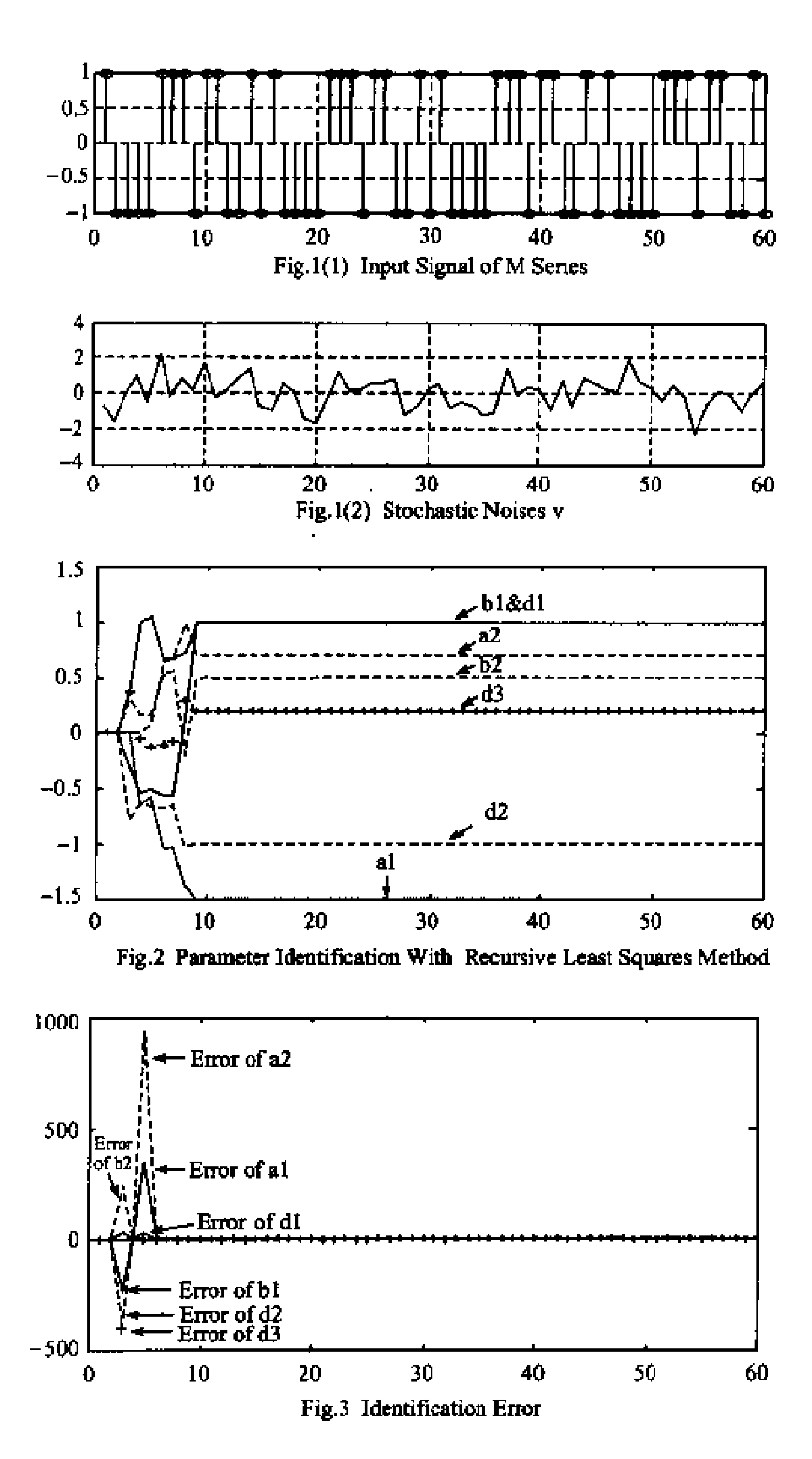


Fig.1(1) Input Signal of M Series

Fig.1(2) Stochastic Noises v

Fig.2 Parameter Identification With Recursive Least Squares Method

Fig.3 Identification Error

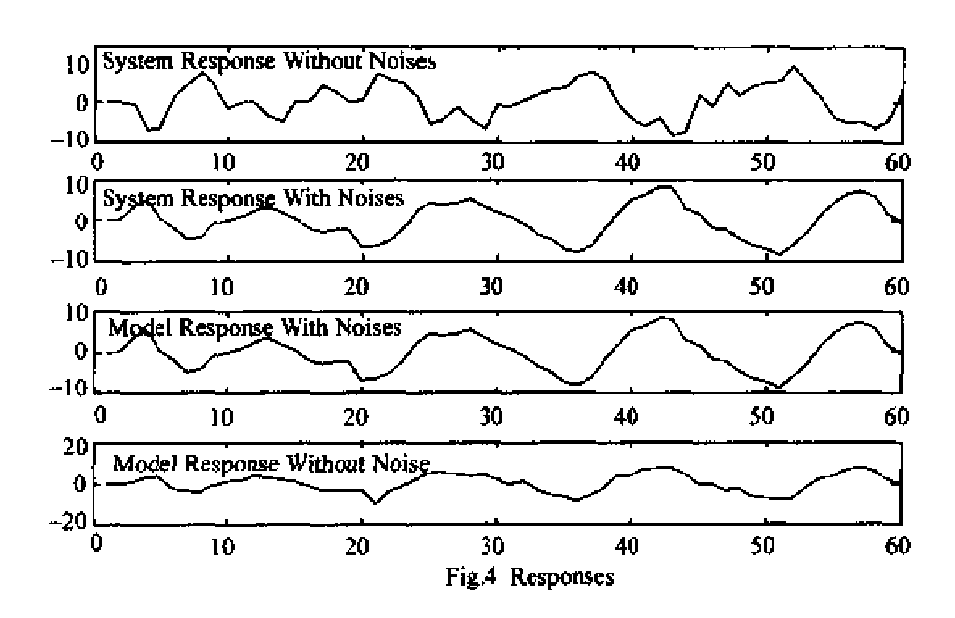


图 3.12 增广最小二乘递推算法辨识仿真

```
%增广最小二乘辨识程序,在光盘中的文件名:FLch3ELSeg4.m
clear
L=60; %4 位移位寄存器产生的 M 序列的周期
y1=1;y2=1;y3=1;y4=0; %4 个移位寄存器的输出初始值
for i=1:L;
    x1=xor(y3,y4); %第一个移位寄存器的输入信号
    x2=y1; %第二个移位寄存器的输入信号
    x3=y2; %第三个移位寄存器的输入信号
    x4=y3; %第四个移位寄存器的输入信号
    y(i)=y4; %第四个移位寄存器的输出信号,M 序列, 幅值 "0" 和 "1",
    if y(i)>0.5,u(i)=-1; %M 序列的值为 "1" 时,辨识的输入信号取 "-1"
    else u(i)=1; %M 序列的值为 "0" 时,辨识的输入信号取 "1"
    end
    y1=x1;y2=x2;y3=x3;y4=x4; %为下一次的输入信号准备
end
figure(1); %画第一个图形
subplot(2,1,1); %画第一个图形的第一个子图
stem(u),grid on %画出 M 序列输入信号
v=randn(1,60); %产生一组 60 个正态分布的随机噪声
subplot(2,1,2); %画第一个图形的第二个子图
plot(v),grid on; %画出随机噪声信号
u ,v %显示输入信号和噪声信号
z=zeros(7,60);zs=zeros(7,60);zm=zeros(7,60);zmd=zeros(7,60); %输出采样矩阵、不考虑噪声时系
%统输出矩阵、不考虑噪声时模型输出矩阵、模型输出矩阵的大小
z(2)=0; z(1)=0; zs(2)=0; zs(1)=0; zm(2)=0; zm(1)=0; zmd(2)=0; zmd(1)=0; %输出采样、不
```

```
%考虑噪声时系统输出、不考虑噪声时模型输出、模型输出的初值
c0=[0.001 0.001 0.001 0.001 0.001 0.001 0.001]'; %直接给出被辨识参数的初始值,即一个充分小的
%实向量
p0=10^6*eye(7,7); %直接给出初始状态 P0,即一个充分大的实数单位矩阵
E=0.00000000005; %相对误差 E=0.000000005
c=[c0,zeros(7,14)]; %被辨识参数矩阵的初始值及大小
e=zeros(7,15); %相对误差的初始值及大小
for k=3:60; %开始求 K
    z(k)=1.5*z(k-1)-0.7*z(k-2)+u(k-1)+0.5*u(k-2)+v(k)-v(k-1)+0.2*v(k-2);
%系统在M序列输入下的输出采样信号
    h1=[-z(k-1),-z(k-2),u(k-1),u(k-2),v(k),v(k-1),v(k-2)]'; %为求 K(k)作准备
    x=h1'*p0*h1+1; x1=inv(x); k1=p0*h1*x1; %K
    d1=z(k)-h1'*c0; c1=c0+k1*d1; %辨识参数 c
    zs(k)=-1.5*z(k-1)+0.7*z(k-2)+u(k-1)+0.5*u(k-2); %系统在 M 序列的输入下的输
%出响应
    zm(k)=[-z(k-1),-z(k-2),u(k-1),u(k-2)]*[c1(1);c1(2);c1(3);c1(4)]; %模型在 M 序
%列的输入下的输出响应
    zmd(k)=h1'*c1; %模型在 M 序列的输入下的输出响应
    e1=c1-c0; e2=e1./c0; %求参数误差的相对变化
    e(:,k)=e2;
    c0=c1; %给下一次用
    c(:,k)=c1; %把递推出的辨识参数 c 的列向量加入辨识参数矩阵
    p1=p0-k1*k1'*[h1'*p0*h1+1]; %find p(k)
    p0=p1; %给下次用
    if e2<=E break; %若收敛情况满足要求,终止计算
    end %判断结束
end %循环结束
c, e, %显示被辨识参数及参数收敛情况
z, zs, zm %显示输出采样值、系统实际输出值、模型输出值
%分离变量
a1=c(1,:); a2=c(2,:); b1=c(3,:); b2=c(4,:); %分离出 a1、a2、b1、b2
d1=c(5,:); d2=c(6,:); d3=c(7,:); %分离出 d1、d2、d3
ea1=e(1,:); ea2=e(2,:); eb1=e(3,:); eb2=e(4,:); %分离出 a1、a2、b1、b2 的收敛情况
ed1=e(5,:); ed2=e(6,:); ed3=e(7,:); %分离出 d1、d2、d3 的收敛情况
figure(2); %画第二个图形
i=1:60;plot(i,a1,'r',i,a2,'r:',i,b1,'b',i,b2,'b:',i,d1,'g',i,d2,'g:',i,d3,'g+') %画出各个被辨识参数
title('Parameter Identification with Recursive Least Squares Method') %标题
figure(3); i=1:60; %画出第三个图形
plot(i,ea1,'r',i,ea2,'r:',i,eb1,'b',i,eb2,'b:',i,ed1,'g',i,ed2,'g:',i,ed2,'r+') %画出各个参数收敛情况
title('Identification Precision') %标题
figure(4); subplot(4,1,1); %画出第四个图形,第一个子图
```

```
i=1:60;plot(i,zs(i),'r') %画出被辨识系统在没有噪声情况下的实际输出响应
subplot(4,1,2); i=1:60;plot(i,z(i),'g') %第二个子图,画出被辨识系统的采样输出响应
subplot(4,1,3); i=1:60;plot(i,zm(i),'b') %第三个子图,画出模型含有噪声的输出响应
subplot(4,1,4); i=1:60;plot(i,zs(i),'b') %第四个子图,画出模型去除噪声后的输出响应
```

程序执行结果:

```
>>
u=
[1,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,1,-1,1,-1,-1,1,-1,-1,1,1,-1,1,1,-1,-1,1,
%-1,1,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,1,-1,1,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,
%1,-1]
v=
[1.1418,1.5519,1.3836,-0.7581,0.4427,0.9111,-1.0741,0.2018,0.7629,-1.2882,-0.9530,
0.7782,-0.0063,0.5245,1.3643,0.4820,-0.7871,0.7520,-0.1669,-0.8162,2.0941,0.0802,
%-0.9373,0.6357,1.6820,0.5936,0.7902,0.1053,-0.1586,0.8709,-0.1948,0.0755,-0.5266,
%-0.6855,-0.2684,-1.1883,0.2486,-0.1025,-0.0410,-2.2476,-0.5108,0.2492 0.3692,
%0.1792,-0.0373,-1.6033,0.3394,-0.1311,0.4852,0.5988,-0.0860,0.3253,-0.3351,-0.3224,
%-0.3824,-0.9534,0.2336,1.2352,-0.5785,-0.5015]

c=
   0.0010  0   0.0010  -0.2789  -0.9025  -0.9390  -0.8620
   0.0010  0   0.0010   0.0010  -0.0734   0.2661  -0.1199
   0.0010  0   0.0592   0.4560   0.5300   0.5293   0.9458
   0.0010  0   0.0572   0.8186   0.8500   0.8639   1.1157
   0.0010  0   0.0796   0.7342   0.5417   0.5516   0.3847
   0.0010  0  -0.0894  -0.5981  -0.3420  -0.4456  -0.4044
   0.0010  0  -0.0655  -0.7795  -0.8572  -0.7412  -0.4506

  -1.1236  -1.5000  -1.5000  -1.5000
   0.3259   0.7000   0.7000   0.7000
   0.5402   1.0000   1.0000   1.0000
   1.2957   0.5000   0.5000   0.5000
   0.2459   1.0000   1.0000   1.0000
  -0.8443  -1.0000  -1.0000  -1.0000
  -0.1974   0.2000   0.2000   0.2000

e=
   0  0         0  -279.9426    2.2355   0.0404  -0.0820
   0  0         0          0  -74.3764   2.6263   0.5495
   0  0   58.2212     6.7007    0.1622  -0.0013   0.7868
   0  0  -58.2212   -15.3052    0.0384   0.0163   0.2915
   0  0  -80.5566   -10.2283   -0.2621   0.0183  -0.3027
   0  0  -90.3556     5.6936   -0.4282   0.3029  -0.0925
   0  0  -66.4754    10.9059    0.0996  -0.1354  -0.3920
```

```
  0.3035    0.3350    0.0000    0.0000
 -3.7189    1.1479   -0.0000    0.0000
 -0.4289    0.8512    0.0000    0.0000
  0.1613   -0.6141   -0.0000   -0.0000
 -0.3607    3.0663    0.0000    0.0000
  1.0880    0.1844   -0.0000    0.0000
 -0.5619   -2.0130    0.0000    0.0000
z=
[0,0, -0.4400, -3.9913, -5.7014, -6.9415, -7.8178, -3.9097, 1.4543, 2.4075, 3.5810, 6.6598,
6.0079,3.5364,2.4376, -0.0964, -2.3472, -2.3178, -4.4100, -6.9915, -6.0233, -5.8181,
% -3.6094,1.7476,5.5068,6.5756,8.0416,6.3933,2.3550, 0.6078, -2.3342, -2.9824, -3.9807,
% -5.5271, -6.6924, -8.7267, -6.5221, -2.5583,2.1343,2.3062,4.1939,6.4870, 6.3125,3.2878,
%0.8703, -3.0262, -2.7133, -3.2428, -3.7807, -4.8137, -6.6618, -5.5921, -2.9025,1.1385,
%3.1125,3.7363,6.0362,6.7499,2.6324,0.0478]
zs=
[0,0, -0.5000, -0.8401,4.1789,4.2583,6.9212,8.3677,1.8920, -5.4182, -2.0932, -2.1863,
% -7.9830, -5.8500, -0.5991, -1.6810,2.3509,2.9533,0.3337,3.4925,5.9002,4.6409,6.0108,
%2.8415, -5.6480, -6.5369, -4.5087, -7.9595, -5.4608, 1.4428,0.2369,4.4268,2.3396,2.3834,
%4.0042,4.6696,8.9054,5.1745,0.7719, -5.4923, -1.4653, -3.1765, -7.2947, -6.4279, -0.0129,
%0.4961,5.6486,1.4516,1.4649,1.9010,3.0741,7.1232,5.2248,1.9393, -4.2395, -3.3718,
% -1.9258, 6.9389, -7.3995,1.2763 ]
zm=
[0,0, -0.4400, -3.9913, -5.7014, -6.9415, -7.8178, -3.9097,1.4543,2.4075,3.5810,6.6598,
%6.0079,3.5364,2.4376, -0.0964, -2.3472, -2.3178, -4.4100, -6.9915, -6.0233, -5.8181,
% -3.6094,1.7476,5.5068,6.5756,8.0416,6.3933,2.3550, 0.6078, -2.3342, -2.9824, -3.9807,
% -5.5271, -6.6924, -8.7267, -6.5221, -2.5583,2.1343,2.3062,4.1939,6.4870,6.3125, 3.2878,
%0.8703, -3.0262, -2.7133, -3.2428, -3.7807, -4.8137, -6.6618, -5.5921, -2.9025,1.1385,
%3.1125,3.7363, 6.0362,6.7499,2.6324,0.0478]
```

表 3.3　增广最小二乘递推算法的辨识结果

参　数	a_1	a_2	b_1	b_2	d_1	d_2	d_3
真　值	-1.5	0.7	1.0	0.5	1.0	-1.0	0.2
估计值	-1.5	0.7	1.0	0.5	1.0	-1.0	0.2

仿真结果表明，递推到第九步时，参数辨识的结果基本达到稳定状态，即 $a_1=-1.5000$，$a_2=0.7000$，$b_1=1.0000$，$b_2=0.5000$，$d_1=1.0000$，$d_2=-1.0000$，$d_3=0.2000$。此时，辨识参数的相对变化量 $E\leqslant 0.000\,000\,005$。与最小二乘递推算法相比，增广最小二乘递推算法虽然考虑了噪声模型，同样具有速度快、辨识结果精确的特点。

3.9 广义最小二乘法

设 SISO 系统采用如下数学模型

$$A(z^{-1})z(k) = B(z^{-1})u(k) + \frac{1}{C(z^{-1})}v(k) \tag{3.81}$$

描述,式中,$u(k)$和 $z(k)$表示系统的输入和输出; $v(k)$是均值为零的不相关的随机噪声;且

$$\begin{cases} A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a} \\ B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b} \\ C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots + c_{n_c} z^{-n_c} \end{cases} \tag{3.82}$$

若假定模型阶次 n_a、n_b 和 n_c 已经确定,则这类问题的辨识可用广义最小二乘法 RGLS(recursive generalized least squares),以便获得参数的无偏一致估计。令

$$\begin{cases} z_f(k) = C(z^{-1})z(k) \\ u_f(k) = C(z^{-1})u(k) \end{cases} \tag{3.83}$$

及

$$\begin{cases} \boldsymbol{\theta} = [a_1, a_2, \cdots, a_{n_a}, b_1, b_2, \cdots, b_{n_b}]^{\mathrm{T}} \\ \boldsymbol{h}_f(k) = [-z_f(k-1), \cdots, -z_f(k-n_a), u_f(k-1), \cdots, u_f(k-n_b)]^{\mathrm{T}} \end{cases} \tag{3.84}$$

将模型(3.81)化为最小二乘格式

$$z_f(k) = \boldsymbol{h}_f^{\mathrm{T}}(k)\boldsymbol{\theta} + v(k) \tag{3.85}$$

由于 $v(k)$是白噪声,所以利用最小二乘法即可获得参数 $\boldsymbol{\theta}$ 的无偏估计。但是数据向量 $\boldsymbol{h}_f(k)$中的变量均需按式(3.83)计算,然而噪声模型 $C(z^{-1})$并不知道。为此需要用迭代的办法来估计 $C(z^{-1})$。令

$$e(k) = \frac{1}{C(z^{-1})}v(k) \tag{3.86}$$

置

$$\begin{cases} \boldsymbol{\theta}_e(k) = [c_1, c_2, \cdots, c_{n_c}]^{\mathrm{T}} \\ \boldsymbol{h}_e(k) = [-e(k-1), \cdots, -e(k-n_c)]^{\mathrm{T}} \end{cases} \tag{3.87}$$

就把噪声模型式(3.86)也化成最小二乘格式

$$e(k) = \boldsymbol{h}_e^{\mathrm{T}}(k)\boldsymbol{\theta}_e + v(k) \tag{3.88}$$

由于上式的噪声已是白噪声,所以再次利用最小二乘法可获得噪声模型参数 $\boldsymbol{\theta}_e$ 的无偏估计。但是数据向量 $\boldsymbol{h}_e(k)$依然包含着不可测的噪声量 $e(k-1)$, …, $e(k-$

n_c)，它可用相应的估计值代替，置

$$\boldsymbol{h}_e(k) = [-\hat{e}(k-1), \cdots, -\hat{e}(k-n_c)]^{\mathrm{T}} \tag{3.89}$$

式中，$k \leqslant 0$，$\hat{e}(k)=0$；当 $k>0$ 时，按下式

$$\hat{e}(k) = z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}} \tag{3.90}$$

计算，式中

$$\boldsymbol{h}(k) = [-z(k-1), \cdots, -z(k-n_a), u(k-1), \cdots, u(k-n_b)]^{\mathrm{T}} \tag{3.91}$$

综上分析，加权广义最小二乘递推算法可归纳成：

$$\begin{cases} \hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{K}_f(k)[z_f(k) - \boldsymbol{h}_f^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1)] \\ \boldsymbol{K}_f(k) = \boldsymbol{P}(k-1)\boldsymbol{h}_f(k)\left[\boldsymbol{h}_f^{\mathrm{T}}(k)\boldsymbol{P}_f(k-1)\boldsymbol{h}_f(k) + \dfrac{1}{\Lambda(k)}\right]^{-1} \\ \boldsymbol{P}_f(k) = [\boldsymbol{I} - \boldsymbol{K}_f(k)\boldsymbol{h}_f^{\mathrm{T}}(k)]\boldsymbol{P}_f(k-1) \\ \hat{\boldsymbol{\theta}}_e(k) = \hat{\boldsymbol{\theta}}_e(k-1) + \boldsymbol{K}_e(k)[\hat{e}(k) - \boldsymbol{h}_e^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}_e(k-1)] \\ \boldsymbol{K}_e(k) = \boldsymbol{P}_e(k-1)\boldsymbol{h}_e(k)\left[\boldsymbol{h}_e^{\mathrm{T}}(k)\boldsymbol{P}_e(k-1)\boldsymbol{h}_e(k) + \dfrac{1}{\Lambda(k)}\right]^{-1} \\ \boldsymbol{P}_e(k) = [\boldsymbol{I} - \boldsymbol{K}_e(k)\boldsymbol{h}_e^{\mathrm{T}}(k)]\boldsymbol{P}_e(k-1) \end{cases} \tag{3.92}$$

当所有采样数据都是等同加权，即 $\Lambda(k)=1$ 时，加权广义最小二乘参数估计递推算法就简化成广义最小二乘参数估计递推算法 RGLS。

广义最小二乘递推算法是一种迭代的算法，其步骤是：

(1) 给定初始条件

$$\begin{cases} \hat{\boldsymbol{\theta}}(0) = \boldsymbol{\varepsilon}(\boldsymbol{\varepsilon}\text{ 为充分小的实向量}) \\ \boldsymbol{P}_f(0) = \alpha^2\boldsymbol{I}(\alpha\text{ 为充分大的数}) \\ \hat{\boldsymbol{\theta}}_e(0) = \boldsymbol{0} \\ \boldsymbol{P}_e(0) = \boldsymbol{I} \end{cases} \tag{3.93}$$

(2) 利用式(3.83)，计算 $z_f(k)$和 $u_f(k)$；

(3) 利用式(3.84)，构造 $\boldsymbol{h}_f(k)$；

(4) 利用式(3.92)前三个式子递推计算$\hat{\boldsymbol{\theta}}(k)$；

(5) 利用式(3.90)计算$\hat{e}(k)$，并根据式(3.89)构造 $\boldsymbol{h}_e(k)$；

(6) 利用式(3.92)后三个式子递推计算$\hat{\boldsymbol{\theta}}_e(k)$；

(7) 返回第(2)步进行迭代计算，直至获得满意的辨识结果。

以上分析表明，广义最小二乘法的基本思想是基于对数据先进行一次滤波预

处理,然后利用普通最小二乘法对滤波后的数据进行辨识。如果滤波模型选择得合适,对数据进行了较好的白色化处理,那么直接利用普通最小二乘法就能获得无偏一致估计。这种滤波模型可以是预先选定的固定模型,也可以是动态变化模型。由于实际问题的复杂性,要选择一个较好的固定模型用于数据的白色化处理一般是比较困难的。广义最小二乘法所用的滤波模型实际上是一种动态模型,在整个迭代过程中不断靠偏差信息来调整这个滤波模型,使它逐渐逼近于一个较好的滤波模型,以便对数据进行较好的白色化处理,使模型参数估计成为无偏一致估计。理论上说,广义最小二乘法所用的动态模型经过几次迭代调整后,便可对数据进行较好的白色化处理。但是,当过程的输出信噪比较大或模型参数比较多时,这种数据白色化处理的可靠性就会下降。此时,准则函数可能会出现多个局部收敛点,辨识结果可能使准则函数收敛于局部极小点而不是全局极小点。这样最终的辨识结果往往也会是有偏的。

例如,考虑仿真对象

$$\begin{cases} z(k) + a_1 z(k-1) + a_2 z(k-2) = b_1 u(k-1) + b_2 u(k-2) + e(k) \\ e(k) + c_1 e(k-1) + c_2 e(k-2) = v(k) \end{cases}$$

利用(3.92)递推算式获得的辨识结果如表3.4所示。若将此例的辨识结果与最小二乘递推算法相比较,可以发现它们的结果基本上是一致的。这是因为本例的仿真对象也采用了图3.6所示的系统,它相当于式(3.81)中的噪声模型 $C(z^{-1})=1$。这种仿真对象利用最小二乘法已可获得无偏一致估计,但广义最小二乘同时又能给出噪声模型的参数估计值。由表3.4知,此例的噪声模型估计为 $\hat{C}(z^{-1})\approx 1$,它和仿真对象的噪声模型是相符的。如果一个系统必须用式(3.81)的模型来描述,且 $C(z^{-1})\neq 1$,这时就不能使用最小二乘法了,必须利用广义最小二乘法,方可获得无偏一致估计。

表3.4　广义最小二乘法辨识结果

参　数	a_1	a_2	b_1	b_2	c_1	c_2
真　值	-1.5	0.7	1.0	0.5	0.0	0.0
估计值	-1.5107	0.7068	1.046	0.4963	0.0114	-0.0034

3.10　多级最小二乘法

考虑 SISO 系统模型

$$A(z^{-1})z(k) = B(z^{-1})u(k) + \frac{1}{C(z^{-1})}v(k) \tag{3.94}$$

当信噪比相对较大时，采用广义最小二乘法可能会出现多个局部收敛点，使辨识结果难以收敛到全局极小点。解决这个问题的方法可用多级最小二乘法 MLS(multi-stage least squares)。一般说来，多级最小二乘法包含三级辨识过程，如图 3.13 所示。

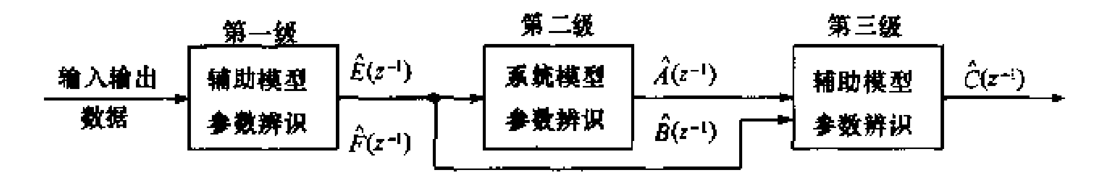


图 3.13 多级最小二乘法

3.10.1 辅助模型参数辨识

辅助模型参数辨识是多级最小二乘辨识的第一级。将模型(3.94)写成

$$A(z^{-1})C(z^{-1})z(k) = B(z^{-1})C(z^{-1})u(k) + v(k) \tag{3.95}$$

令

$$\begin{cases} E(z^{-1}) = A(z^{-1})C(z^{-1}) = 1 + e_1 z^{-1} + \cdots + e_{n_a+n_c} z^{-(n_a+n_c)} \\ F(z^{-1}) = B(z^{-1})C(z^{-1}) = f_1 z^{-1} + \cdots + f_{n_b+n_c} z^{-(n_b+n_c)} \end{cases} \tag{3.96}$$

则模型(3.95)写成

$$E(z^{-1})z(k) = F(z^{-1})u(k) + v(k) \tag{3.97}$$

上式称作辅助模型。设

$$\begin{cases} \boldsymbol{\theta}_1 = [e_1, e_2, \cdots, e_{n_a+n_c}, f_1, f_2, \cdots, f_{n_b+n_c}]^{\mathrm{T}} \\ \boldsymbol{h}_1(k) = [-z(k-1), \cdots, -z(k-n_a-n_c), u(k-1), \cdots, u(k-n_b-n_c)]^{\mathrm{T}} \end{cases} \tag{3.98}$$

则模型(3.97)化成最小二乘格式

$$z_1(k) = \boldsymbol{h}_1^{\mathrm{T}}(k)\boldsymbol{\theta}_1 + v(k) \tag{3.99}$$

由于 $v(k)$是白噪声，故利用最小二乘法可获得辅助模型的参数无偏一致估计$\hat{\boldsymbol{\theta}}_1$ 为

$$\hat{\boldsymbol{\theta}}_1 = (\boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{H}_1)^{-1}\boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{z}_1 \tag{3.100}$$

式中

$$\begin{cases} \boldsymbol{z}_1 = [z(1), z(2), \cdots, z(L)]^{\mathrm{T}} \\ \boldsymbol{H}_1 = [\boldsymbol{h}_1^{\mathrm{T}}(1), \boldsymbol{h}_1^{\mathrm{T}}(2), \cdots, \boldsymbol{h}_1^{\mathrm{T}}(L)] \end{cases} \tag{3.101}$$

3.10.2 系统模型参数辨识

系统模型参数辨识是多级最小二乘辨识的第二级。根据式(3.96)，有

$$A(z^{-1})F(z^{-1}) = B(z^{-1})E(z^{-1}) \tag{3.102}$$

定义

$$\boldsymbol{\theta}_2 = [a_1, a_2, \cdots, a_{n_a}, b_1, b_2, \cdots, b_{n_b}]^{\mathrm{T}} \tag{3.103}$$

比较式(3.102)两边 z^{-1}同幂项的系数，可得

$$z_2 = \boldsymbol{H}_2\boldsymbol{\theta}_2 \tag{3.104}$$

式中

$$\begin{cases} z_2 = [f_1, f_2, \cdots, f_{n_b+n_c}, \underbrace{0, \cdots, 0}_{n_a}]^{\mathrm{T}} \\ \boldsymbol{H}_2 = \left[\begin{array}{ccc:ccc} 0 & \cdots & 0 & 1 & \cdots & 0 \\ -f_1 & \ddots & 0 & e_1 & \ddots & 1 \\ -f_2 & \ddots & -f_1 & e_2 & \ddots & e_1 \\ \vdots & \ddots & -f_2 & \vdots & \ddots & e_2 \\ -f_{n_b+n_c} & & \vdots & e_{n_a+n_c} & & \vdots \\ 0 & \ddots & -f_{n_b+n_c} & 0 & \cdots & e_{n_a+n_c} \\ & n_a & & & n_b & \end{array}\right] \Bigg\} n_a + n_b + n_c \end{cases} \tag{3.105}$$

若上式 z_2和 $\boldsymbol{H}_2$ 中的元素 f_i 和 e_i 分别用它们的估计值 $\hat{f}_i$ 和 $\hat{e}_i$ 代替，则式(3.104)可写成

$$\hat{z}_2(k) = \hat{\boldsymbol{H}}_2\boldsymbol{\theta}_2 + \boldsymbol{n}_2 \tag{3.106}$$

式中，$\boldsymbol{n}_2$是 z_2和 $\boldsymbol{H}_2$ 中的元素用对应的估计值代替后所引起的误差项。视 $\boldsymbol{n}_2$为白噪声，则可利用最小二乘法获得系统模型的参数估计值$\hat{\boldsymbol{\theta}}_2$

$$\hat{\boldsymbol{\theta}}_2 = (\hat{\boldsymbol{H}}_2^{\mathrm{T}}\hat{\boldsymbol{H}}_2)^{-1}\hat{\boldsymbol{H}}_2^{\mathrm{T}}\hat{z}_2 \tag{3.107}$$

式中，$\hat{\boldsymbol{H}}_2$ 和$\hat{z}_2$ 表示(3.105)中的元素均用估计值代替。

3.10.3 噪声模型参数辨识

噪声模型参数辨识是多级最小二乘辨识的第三级。根据式(3.96)，有

$$\begin{cases} E(z^{-1}) = A(z^{-1})C(z^{-1}) \\ F(z^{-1}) = B(z^{-1})C(z^{-1}) \end{cases} \tag{3.108}$$

定义

$$\boldsymbol{\theta}_3 = [c_1, c_2, \cdots, c_{n_c}]^{\mathrm{T}} \tag{3.109}$$

比较式(3.108)两边 z^{-1}同幂项的系数，可得

$$z_3 = \boldsymbol{H}_3\boldsymbol{\theta}_3 \tag{3.110}$$

式中

$$\begin{cases} z_3 = [e_1 - a_1, \cdots, e_{n_a} - a_{n_a}, e_{n_a+1}, \cdots, e_{n_a+n_c}, \\ \quad f_2 - b_2, \cdots, f_{n_b} - b_{n_b}, b_{n_b+1}, \cdots, b_{n_b+n_c}]^{\mathrm{T}} \\ \boldsymbol{H}_3 = \begin{bmatrix} 1 & \cdots & 0 \\ a_1 & \ddots & \vdots \\ \vdots & \ddots & 1 \\ a_{n_a} & & a_1 \\ 0 & \ddots & \vdots \\ b_1 & \ddots & a_{n_a} \\ \vdots & \ddots & 0 \\ b_{n_b} & & b_1 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b_{n_b} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}1\\1\\1\end{matrix}}\right\} n_c \\ \left.\vphantom{\begin{matrix}1\\1\\1\end{matrix}}\right\} n_a \\ \left.\vphantom{\begin{matrix}1\\1\end{matrix}}\right\} n_c-1 \\ \left.\vphantom{\begin{matrix}1\\1\\1\end{matrix}}\right\} n_b \end{matrix} \end{cases} \tag{3.111}$$

若上式 z_3和 $\boldsymbol{H}_3$ 中的元素 a_i、b_i、f_i 和 e_i 分别用对应的估计值代替，则式(3.109)可写成

$$\hat{z}_3(k) = \hat{\boldsymbol{H}}_3\boldsymbol{\theta}_3 + \boldsymbol{n}_3 \tag{3.112}$$

式中，$\boldsymbol{n}_3$是 z_3和 $\boldsymbol{H}_3$ 中的元素用对应的估计值代替后所引起的误差项。视 $\boldsymbol{n}_3$为白噪声，则可利用最小二乘法获得系统模型的参数估计值$\hat{\boldsymbol{\theta}}_3$

$$\hat{\boldsymbol{\theta}}_3 = (\boldsymbol{H}_3^{\mathrm{T}}\boldsymbol{H}_3)^{-1}\boldsymbol{H}_3^{\mathrm{T}}\hat{z}_3 \tag{3.113}$$

式中，$\hat{\boldsymbol{H}}_3$ 和$\hat{z}_3$ 表示式(3.111)中相应的元素均用估计值代替。

综上分析，式(3.100)、(3.107)和(3.113)组成了多级最小二乘法，利用输入输出数据，可以分别求得辅助模型、系统模型和噪声模型的参数估计值。在高信噪比的情况下，多级最小二乘法将明显优于广义最小二乘法，广义最小二乘法可能出现多个收敛点，而多级最小二乘法的收敛点是惟一的。

例如，考虑仿真对象

$$\begin{cases} z(k) + 0.9z(k-1) + 0.15z(k-2) + 0.02z(k-3) \\ \qquad = 0.7u(k-1) - 1.5u(k-2) + e(k) \\ e(k) + 1.0e(k-1) + 0.41e(k-2) = \lambda v(k) \end{cases}$$

式中，$u(k)$和 $z(k)$是输入输出数据；$v(k)$是零均值、方差为 1 的不相关的随机噪声；$u(k)$采用与 $e(k)$不相关的随机序列；控制 λ 值，使信噪比 $\eta=92\%$。模型结构选用

$$A(z^{-1})z(k) = B(z^{-1})u(k) + \frac{1}{C(z^{-1})}v(k)$$

式中

$$\begin{cases} A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} \\ B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} \\ C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} \end{cases}$$

当数据长度取 $L=450$ 时，利用多级最小二乘法的辨识结果如表 3.5 所示。此例若用广义最小二乘法，在高信噪比情况下，将无法获得参数的无偏估计值。可见，当信噪比相对较大时，多级最小二乘法比广义最小二乘法更有使用价值。

表 3.5　多级最小二乘法辨识结果

参　数	a_1	a_2	a_3	b_1	b_2	c_1	c_2
真　值	0.9	0.15	0.02	0.7	−1.5	1.0	0.41
估计值	0.89558	0.15798	0.01463	0.69688	−1.4970	1.01149	0.40463

3.11　小　　结

最小二乘法是 1795 年高斯在他著名的星体运动轨道预报研究工作中提出的。此后，它就成了估计理论的奠基石。最小二乘法思想是使各次实际观测值和计算值之间差值的平方乘以度量其精确度的数值以后的和为最小。由于最小二乘法原理简单，编制程序也不困难，所以它颇受人们重视，应用相当广泛。

最小二乘法的基本结果有两种形式：一种是经典的一次完成算法；另一种是现代的递推算法。最小二乘一次完成算法比较适合理论研究，但编制程序时占用的存储空间较多，计算量较大，所以多用于离线系统辨识；最小二乘递推算法的基本思想是新的估计值等于前一次的估计值加上修正项，这样不仅可以减少计算量和存储量，而且能实现系统的在线辨识。

增广最小二乘法扩充了最小二乘法的参数向量和数据向量，把噪声模型的辨识同时考虑了进去。增广最小二乘法又是一种近似的极大似然法。

广义最小二乘法的基本思想是基于对数据先进行一次滤波预处理，然后利用普通最小二乘法对滤波后的数据进行辨识。广义最小二乘法所用的滤波模型实际上是一种动态模型，在整个迭代过程中靠偏差信息来不断地调整滤波模型，使之逐渐逼近于一个较好的滤波模型，以便对数据进行较好的白色化处理，使模型参数估

计成为无偏一致估计。

多级最小二乘法是利用输入输出数据,分别求出辅助模型、系统模型和噪声模型的参数估计值。在高信噪比的情况下,多级最小二乘法将明显优于广义最小二乘法,因为多级最小二乘法的收敛点是惟一的。

本章的相应部分,分别给出了 2 个最小二乘一次完成算法的 MATLAB 6.0 仿真实例、1 个最小二乘递推算法的 MATLAB 6.0 仿真实例和 1 个增广最小二乘辨识的 MATLAB 6.0 仿真实例。

多变量线性系统的辨识参看文献[42]的第 15 章及文献[45]的第 6 章。

习　题

1. 什么是最小二乘参数辨识问题,简述其基本原理。
2. 简述在最小二乘参数估计问题中引入加权因子的作用。
3. 1946～1956 年美国的钢年产量如表 2.6(单位为百万吨):

表 2.6

年	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956
产量	66.6	84.9	88.6	78.0	96.8	105.2	93.2	111.6	88.3	117.0	115.2

若钢年产量用如下两种模型描述:① $s=\theta_0+\theta_1 t$;② $s=\theta_0+\theta_1 t+\theta_2 t^2$。式中,$s$ 表示钢年产量;t 表示年代。试用最小二乘的一次完成算法确定这两种模型的参数 θ_0、θ_1、θ_2。

4. 考虑图 3.6 所示的仿真对象,其中,$v(k)$是服从 $\mathcal{N}(0,1)$分布的不相关随机噪声。且

$$G(z^{-1})=B(z^{-1})/A(z^{-1}),N(z^{-1})=D(z^{-1})/C(z^{-1})$$

$$\begin{cases}A(z^{-1})=1-2.5a_1 z^{-1}+1.7z^{-2}=C(z^{-1})\\B(z^{-1})=1.6z^{-1}+0.8z^{-2}\\D(z^{-1})=1\end{cases}$$

若仿真对象选择如下的模型结构

$$z(k)+a_1 z(k-1)+a_2 z(k-2)=b_1 u(k-1)+b_2 u(k-2)+v(k)$$

试用递推最小二乘算法确定模型参数 a_1、a_2、b_1、b_2。

5. 设某物理量 y 与 x_1、x_2 和 x_3 满足关系:$y=\theta_1 x_1+\theta_2 x_2+\theta_3 x_3$。实验获得一批数据如表 2.7,试用最小二乘法确定模型参数 θ_1、θ_2 和 θ_3。

表 2.7

x_1	0.62	0.40	0.42	0.82	0.66	0.72	0.38	0.52	0.45	0.69	0.55	0.36
x_2	12.0	14.2	14.6	12.1	10.8	8.2	13.0	10.5	8.8	17.0	14.2	12.8
x_3	5.2	6.1	7.32	8.3	5.1	7.9	4.2	8.0	3.9	5.5	3.8	6.2
y	51.6	49.9	48.5	50.6	49.7	48.8	42.6	45.9	37.8	64.8	53.4	45.3

第 4 章　梯度校正参数辨识

梯度校正参数辨识方法的基本思想是沿着准则函数的负梯度方向，逐渐修正模型参数的估计值，直至准则函数达到最小值。从第 3 章的最小二乘参数辨识中可知，递推算法的结构为：新的参数估计值等于老的参数估计值加上增益矩阵与新息的乘积。梯度校正参数辨识方法具有相同的结构，但其基本原理却完全不同于最小二乘参数辨识。本章将着重研究四个问题：确定性问题的梯度校正参数辨识，它是研究随机性问题的基础；随机性问题的梯度校正参数辨识；梯度校正法在动态系统辨识中的应用；随机逼近法。

4.1　确定性问题的梯度校正参数辨识方法

4.1.1　确定性梯度校正辨识公式的推导

设系统的输出 $y(t)$ 是参数 $\theta_1,\theta_2,\cdots,\theta_N$ 的线性组合

$$y(t) = h_1(t)\theta_1 + h_2(t)\theta_2 + \cdots + h_N(t)\theta_N \tag{4.1}$$

如果输出 $y(t)$ 和输入 $h_1(t),h_2(t),\cdots,h_N(t)$ 是可以准确测量的，则系统(4.1)称作确定性系统，如图 4.1 所示。



图 4.1　确定性系统

置

$$\begin{cases} \boldsymbol{h}(t) = [h_1(t),h_2(t),\cdots,h_N(t)]^{\mathrm{T}} \\ \boldsymbol{\theta} = [\theta_1,\theta_2,\cdots,\theta_N]^{\mathrm{T}} \end{cases} \tag{4.2}$$

若系统参数的真值记作 $\boldsymbol{\theta}_0$，则有

$$y(t) = \boldsymbol{h}^{\mathrm{T}}(t)\boldsymbol{\theta}_0 \tag{4.3}$$

在离散点又可以写成

$$y(k) = \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}_0 \tag{4.4}$$

式中

$$\boldsymbol{h}(k) = [h_1(k),h_2(k),\cdots,h_N(k)]^{\mathrm{T}} \tag{4.5}$$

例如，用差分方程描述的确定性系统

$$y(k)+a_1y(k-1)+\cdots+a_ny(k-n)=b_1u(k-1)+\cdots+b_nu(k-n) \tag{4.6}$$

很容易化为式(4.4)的形式,其中

$$\begin{cases}\boldsymbol{h}(k)=[-y(k-1),\ \cdots,\ -y(k-n),u(k-1),\ \cdots,u(k-n)]^{\mathrm{T}}\\ \boldsymbol{\theta}=[a_1,a_2,\ \cdots,a_n,b_1,b_2,\ \cdots,b_n]^{\mathrm{T}}\end{cases} \tag{4.7}$$

现在的问题就是如何利用输入输出数据 $\boldsymbol{h}(k)$和 $y(k)$来确定参数 $\boldsymbol{\theta}$ 在k 时刻的估计值$\hat{\boldsymbol{\theta}}(k)$,使准则函数

$$J(\boldsymbol{\theta})\Big|_{\hat{\boldsymbol{\theta}}(k)}=\frac{1}{2}\varepsilon^2(\boldsymbol{\theta},k)\Big|_{\hat{\boldsymbol{\theta}}(k)}=\min \tag{4.8}$$

式中

$$\varepsilon(\boldsymbol{\theta},k)=y(k)-\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta} \tag{4.9}$$

梯度校正法,即最速下降法,可以解决上述问题。具体做法就是沿着 $J(\boldsymbol{\theta})$的负梯度方向不断修正$\hat{\boldsymbol{\theta}}(k)$值,直至 $J(\boldsymbol{\theta})$达到最小。梯度校正法的数学表达式为

$$\hat{\boldsymbol{\theta}}(k+1)=\hat{\boldsymbol{\theta}}(k)-\boldsymbol{R}(k)\,\underset{\boldsymbol{\theta}}{\mathbf{grad}}[J(\boldsymbol{\theta})]\Big|_{\hat{\boldsymbol{\theta}}(k)} \tag{4.10}$$

式中,$\boldsymbol{R}(k)$是 N 维对称矩阵,称作加权阵;$\underset{\boldsymbol{\theta}}{\mathbf{grad}}[J(\boldsymbol{\theta})]$表示准则函数 $J(\boldsymbol{\theta})$关于 $\boldsymbol{\theta}$ 的梯度。当准则函数 $J(\boldsymbol{\theta})$取式(4.8)时,有

$$\begin{aligned}\underset{\boldsymbol{\theta}}{\mathbf{grad}}[J(\boldsymbol{\theta})]\Big|_{\hat{\boldsymbol{\theta}}(k)}&=\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\left[\frac{1}{2}\varepsilon^2(\boldsymbol{\theta},k)\right]\Bigg|_{\hat{\boldsymbol{\theta}}(k)}\\&=-\varepsilon(\hat{\boldsymbol{\theta}}(k),k)\boldsymbol{h}(k)=-[y(k)-\boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k)]\boldsymbol{h}(k)\end{aligned} \tag{4.11}$$

则式(4.10)可写成

$$\hat{\boldsymbol{\theta}}(k+1)=\hat{\boldsymbol{\theta}}(k)+\boldsymbol{R}(k)\boldsymbol{h}(k)[y(k)-\boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k)] \tag{4.12}$$

此式就是确定性问题的梯度校正参数估计递推公式,其权矩阵 $\boldsymbol{R}(k)$的选择是非常关键的。

4.1.2 权矩阵的选择

权矩阵 $\boldsymbol{R}(k)$的作用是用来控制各输入分量对参数估计值的影响程度。由式(4.12)知,输入数据向量 $\boldsymbol{h}(k)$的各分量 $h_i(k)$将直接影响参数估计值。设权矩阵 $\boldsymbol{R}(k)$具有如下形式

$$\boldsymbol{R}(k)=c(k)\mathbf{diag}[\Lambda_1(k),\Lambda_2(k),\ \cdots,\Lambda_N(k)] \tag{4.13}$$

只要选择 $\Lambda_i(k)$,就能控制各输入分量 $h_i(k)$对参数估计的影响。例如,如果选择

$$\Lambda_i(k)=\mu^i,0<\mu<1;i=1,2,\cdots,N \tag{4.14}$$

意味着输入分量 $h_{i+1}(k)$ 对参数估计值的影响较 $h_i(k)$ 弱。显然,这种情况 $h_N(k)$ 对参数估计值的影响最小。如果选择

$$\mathbf{diag}[\Lambda_1(k),\Lambda_2(k),\cdots,\Lambda_N(k)]=\boldsymbol{I} \tag{4.15}$$

意味着输入分量的加权值相同,它们对参数估计值的影响是一样的。至于如何合理地选择加权值 $\Lambda_i(k)$,这需要视具体问题而定。权矩阵 $\boldsymbol{R}(k)$ 中的标量 $c(k)$ 由定理 4.1 给出。

定理 4.1 确定性问题的梯度校正参数估计递推公式为

$$\hat{\boldsymbol{\theta}}(k+1)=\hat{\boldsymbol{\theta}}(k)+\boldsymbol{R}(k)\boldsymbol{h}(k)\varepsilon(k) \tag{4.16a}$$

式中

$$\varepsilon(k)=y(k)-\boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k) \tag{4.16b}$$

且权矩阵 $\boldsymbol{R}(k)$ 选如下形式

$$\boldsymbol{R}(k)=c(k)\mathbf{diag}[\Lambda_1(k),\Lambda_2(k),\cdots,\Lambda_N(k)] \tag{4.17}$$

如果权矩阵 $\boldsymbol{R}(k)$ 的元素满足下列条件

(1) $0<\Lambda_L\leqslant\Lambda_i(k)\leqslant\Lambda_H(i=1,2,\cdots,N)$,$\Lambda_H$ 和 Λ_L 为确定的上、下界值;

(2) N 个 $\Lambda_i(k)$ 中至少存在一个 $\Lambda_m(k)$,使得

$$\frac{\Lambda_m(k)-\Lambda_m(k+1)}{\Lambda_m(k)}\geqslant\frac{\Lambda_i(k)-\Lambda_i(k+1)}{\Lambda_i(k)} \tag{4.18}$$

或

$$\frac{\Lambda_m(k+1)}{\Lambda_m(k)}\leqslant\frac{\Lambda_i(k+1)}{\Lambda_i(k)} \tag{4.19}$$

(3) $0<c(k)<\dfrac{2}{\sum\limits_{i=1}^{N}\Lambda_i(k)h_i^2(k)}$

(4) $\tilde{\boldsymbol{\theta}}(k)$ 与 $\boldsymbol{h}(k)$ 不相交,其中 $\tilde{\boldsymbol{\theta}}(k)=\boldsymbol{\theta}_0-\hat{\boldsymbol{\theta}}(k)$,那么不管参数估计的初始值如何选择,参数估计值 $\hat{\boldsymbol{\theta}}(k)$ 总是大范围一致渐近收敛的,即有

$$\lim_{k\to 0}\hat{\boldsymbol{\theta}}(k)=\boldsymbol{\theta}_0 \tag{4.20}$$

定理 4.1 的条件(1)规定加权值 $\Lambda_i(k)$ 必须是大于零且有界的实数;条件(2)是推导条件(3)的前提;条件(3)是保证 $\hat{\boldsymbol{\theta}}(k)$ 大范围一致渐进收敛的条件。定理 4.1 的证明思路如下:

根据式(4.16)及参数估计偏差 $\tilde{\boldsymbol{\theta}}(k)$ 的定义,可得

$$\tilde{\boldsymbol{\theta}}(k+1)=[\boldsymbol{I}-\boldsymbol{R}(k)\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)]\tilde{\boldsymbol{\theta}}(k) \tag{4.21}$$

显然,该式是 $\tilde{\boldsymbol{\theta}}(k)$ 的离散时间运动方程。

设标量函数

$$V[\tilde{\boldsymbol{\theta}}(k),k] = \Lambda_m(k)\sum_{i=1}^{N}\frac{\tilde{\theta}_i^2(k)}{\Lambda_i(k)} \tag{4.22}$$

式中,$\tilde{\theta}_i(k)=\theta_i-\hat{\theta}_i(k)$,$\theta_i$ 是真实参数向量 $\boldsymbol{\theta}_0$ 的第 i 个元素;且 $\Lambda_m(k)$满足条件(2)。可以证明 $V[\tilde{\boldsymbol{\theta}}(k),k]$是方程(4.21)的 Lyapunov 函数。应用 Lyapunov 主稳定性定理可以证明,当条件(2)和(3)成立时,方程(4.21)在平衡状态 $\tilde{\boldsymbol{\theta}}(k)=\mathbf{0}$ 点上是大范围一致渐进稳定的。

Lyapunov 主稳定性要求:

(1) $V[\tilde{\boldsymbol{\theta}}(k),k]>0$,对所有的 $\tilde{\boldsymbol{\theta}}(k)\neq\mathbf{0}$;

(2) $V[\tilde{\boldsymbol{\theta}}(k),k]=0$;有 $\tilde{\boldsymbol{\theta}}(k)=\mathbf{0}$;

(3) 当$\|\tilde{\boldsymbol{\theta}}(k)\|\to\infty$时,有 $V[\tilde{\boldsymbol{\theta}}(k),k]\to\infty$;

(4) $\Delta V[\tilde{\boldsymbol{\theta}},k]<0$,对所有的 $\tilde{\boldsymbol{\theta}}(k)\neq\mathbf{0}$,式中

$$\Delta V[\tilde{\boldsymbol{\theta}},k] = V[\tilde{\boldsymbol{\theta}}(k+1),k+1]-V[\tilde{\boldsymbol{\theta}}(k),k] \tag{4.23}$$

由式(4.18)和(4.22)知,(1)、(2)和(3)一定满足,下面证明(4)。

令

$$\Delta V_m[\tilde{\boldsymbol{\theta}},k] = \frac{V[\tilde{\boldsymbol{\theta}}(k+1),k+1]}{\Lambda_m(k+1)}-\frac{V[\tilde{\boldsymbol{\theta}}(k),k]}{\Lambda_m(k)} \tag{4.24}$$

根据式(4.22),可得

$$\begin{aligned}\Delta V_m[\tilde{\boldsymbol{\theta}},k] &= \sum_{i=1}^{N}\left[\frac{\tilde{\theta}_i^2(k+1)-\tilde{\theta}_i^2(k)}{\Lambda_i(k)}\right]+\sum_{i=1}^{N}\tilde{\theta}_i^2(k+1)\frac{\Lambda_i(k)-\Lambda_i(k+1)}{\Lambda_i(k)\Lambda_i(k+1)}\\ &= Q+\sum_{i=1}^{N}\tilde{\theta}_i^2(k+1)\frac{\Lambda_i(k)-\Lambda_i(k+1)}{\Lambda_i(k)\Lambda_i(k+1)}\end{aligned} \tag{4.25}$$

式中

$$\begin{aligned}Q &= \sum_{i=1}^{N}\left[\frac{\tilde{\theta}_i^2(k+1)-\tilde{\theta}_i^2(k)}{\Lambda_i(k)}\right]\\ &= \sum_{i=1}^{N}\frac{1}{\Lambda_i(k)}[\hat{\theta}_i(k+1)-\hat{\theta}_i(k)][\hat{\theta}_i(k+1)+\hat{\theta}_i(k)-2\,\hat{\theta}_i]\end{aligned} \tag{4.26}$$

利用式(4.16),上式可化成

$$\begin{aligned}Q &= c(k)\varepsilon(k)\sum_{i=1}^{N}h_i(k)[c(k)\Lambda_i(k)h_i(k)\varepsilon(k)-2\tilde{\theta}_i(k)]\\ &= c(k)\varepsilon^2(k)[c(k)\sum_{i=1}^{N}\Lambda_i(k)h_i^2(k)-2]\end{aligned} \tag{4.27}$$

同理,根据定理 4.1 所给的条件(2)及式(4.22),式(4.25)可进一步写成

$$\Delta V_m(\tilde{\boldsymbol{\theta}},k)\leqslant Q+\frac{\Lambda_m(k)-\Lambda_m(k+1)}{\Lambda_m(k)}\sum_{i=1}^{N}\frac{\tilde{\theta}_i^2(k+1)}{\Lambda_i(k+1)}$$

$$= Q + \frac{\Lambda_m(k) - \Lambda_m(k+1)}{\Lambda_m(k)\Lambda_m(k+1)} V[\tilde{\boldsymbol{\theta}}(k+1), k+1]$$

$$= Q + \frac{V[\tilde{\boldsymbol{\theta}}(k+1), k+1]}{\Lambda_m(k)} - \frac{V[\tilde{\boldsymbol{\theta}}(k+1), k+1]}{\Lambda_m(k)} \tag{4.28}$$

注意到 $\Delta V_m[\tilde{\boldsymbol{\theta}}, k]$ 和 $\Delta V[\tilde{\boldsymbol{\theta}}, k]$ 的定义可得

$$\Delta V[\tilde{\boldsymbol{\theta}}, k] \leqslant Q\Lambda_m(k) \tag{4.29}$$

由于 $\Lambda_m(k) > 0$，所以为了使 $\Delta V[\tilde{\boldsymbol{\theta}}, k] < 0$，必须 $Q < 0$，即要求

$$c(k)\varepsilon^2(k)\left[c(k)\sum_{i=1}^{N}\Lambda_i(k)h_i^2(k) - 2\right] < 0 \tag{4.30}$$

由定理 4.1 所给的条件(4)知，$\tilde{\boldsymbol{\theta}}(k)$ 与 $\boldsymbol{h}(k)$ 不正定，故 $\varepsilon(k) = \boldsymbol{h}^{\mathrm{T}}(k)\tilde{\boldsymbol{\theta}}(k) \neq 0$。因此不等式(4.30)的解为

$$0 < c(k) < \frac{2}{\sum_{i=1}^{N}\Lambda_i(k)h_i^2(k)} \tag{4.31}$$

综上所述，只要定理 4.1 的条件得到满足，必有 $\Delta V[\tilde{\boldsymbol{\theta}}, k] < 0$，于是应用 Lyapunov 主稳定性定理可以判断 $\tilde{\boldsymbol{\theta}}(k)$ 是渐进一致收敛的。也就是说，如果 $\tilde{\boldsymbol{\theta}}(k)$ 与 $\boldsymbol{h}(k)$ 不正定，权矩阵 $\boldsymbol{R}(k)$ 选用

$$\begin{cases} \boldsymbol{R}(k) = \dfrac{c}{\sum_{i=1}^{N}\Lambda_i(k)h_i^2(k)}\,\mathbf{diag}[\Lambda_1(k), \Lambda_2(k), \cdots, \Lambda_N(k)] \\ 0 < c < 2 \end{cases} \tag{4.32}$$

则方程(4.21)在 $\tilde{\boldsymbol{\theta}}(k) = \boldsymbol{0}$ 点上是大范围一致渐进稳定的，即有

$$\lim_{k\to\infty}\hat{\boldsymbol{\theta}}(k) = \boldsymbol{\theta}_0 \tag{4.33}$$

例如，I. Nagumo 和 A. Noda 在一个实验研究中曾选择 $\Lambda_i(k) = 1\ (i = 1, 2, \cdots, N)$，即取

$$\boldsymbol{R}(k) = \frac{c\boldsymbol{I}}{\|\boldsymbol{h}(k)\|^2} \tag{4.34}$$

实验证实参数估计值 $\hat{\boldsymbol{\theta}}(k)$ 能较好的收敛于真值 $\boldsymbol{\theta}_0$。

如果 $\tilde{\boldsymbol{\theta}}(k)$ 与 $\boldsymbol{h}(k)$ 正交或时间 k 大于某值后，出现 $\tilde{\boldsymbol{\theta}}(k)$ 与 $\boldsymbol{h}(k)$ 正交，则方程(4.21)不再是大范围一致渐进稳定。也就是说，当 $k\to\infty$ 时，$\tilde{\boldsymbol{\theta}}(k)$ 不会趋于零，而且 $\|\tilde{\boldsymbol{\theta}}(k)\| \leqslant \epsilon$，$\epsilon$ 是大于零的实数。如果输入量 $\boldsymbol{h}(k)$ 每个分量所含的独立频率分量不少于 $(N+1)/2$ 种，则方程(4.21)还是大范围一致渐进稳定。可见，在实际应用中，定理 4.1 的条件(4)必须给与充分注意。

定理 4.1 的条件(3)还告诉我们，权矩阵 $\boldsymbol{R}(k)$ 中的标量 $c(k)$ 存在一个选择

范围。那么在这个范围内是否存在一个最佳值? Lyapunov 稳定性定理指出,如果 $\Delta V[\tilde{\boldsymbol{\theta}},k]$值越负,$\tilde{\boldsymbol{\theta}}(k)$收敛于 $\boldsymbol{\theta}_0$ 的速度越快。如果存在一个 $c^*(k)$,使得 $|\Delta V[\tilde{\boldsymbol{\theta}},k]|=\max$,则这个 $c^*(k)$就是所要找的最佳选择值。根据式(4.29),为了使$|\Delta V[\tilde{\boldsymbol{\theta}},k]|=\max$,$c^*(k)$值必须使$|Q|=\max$。为此,由

$$\left.\frac{\partial Q}{\partial c(k)}\right|_{c^*(k)}=0 \tag{4.35}$$

可得

$$2c^*(k)\varepsilon^2(k)\sum_{i=1}^{N}\Lambda_i(k)h_i^2(k)-2\varepsilon^2(k)=0 \tag{4.36}$$

由于 $\tilde{\boldsymbol{\theta}}(k)$与 $\boldsymbol{h}(k)$不正交,所以 $\varepsilon^2(k)\neq 0$;且不难验证

$$\left.\frac{\partial^2 Q}{\partial c^2(k)}\right|_{c^*(k)}>0 \tag{4.37}$$

故得 $c(k)$的最佳选择值为

$$c^*(k)=\frac{1}{\sum_{i=1}^{N}\Lambda_i(k)h_i^2(k)} \tag{4.38}$$

则最佳权矩阵可写成

$$\boldsymbol{R}^*(k)=\frac{1}{\sum_{i=1}^{N}\Lambda_i(k)h_i^2(k)}\mathbf{diag}[\Lambda_1(k),\Lambda_2(k),\cdots,\Lambda_N(k)] \tag{4.39}$$

$\boldsymbol{R}^*(k)$通常称作 Lyapunov 最佳权矩阵。如果式(4.16)算法中的权矩阵选用 $\boldsymbol{R}^*(k)$,则参数估计值将以最快的速度收敛于真值 $\boldsymbol{\theta}_0$。

4.2 脉冲响应梯度校正辨识的 MATLAB 仿真

考虑图 4.2 表示的线性时不变 SISO 系统。根据卷积定理,系统的输出 $y(k)$与输入序列 $u(k-1),u(k-2),\cdots,u(k-N)$的关系可表示成

$$y(k)=\sum_{i=1}^{N}g_iu(k-i) \tag{4.40}$$

式中,$g_1,g_2,\cdots,g_N$ 组成系统的脉冲响应。系统在伪随机码输入作用下的输出响应如表 4.1 所示。

图 4.2 单输入单输出系统

表 4.1　输入输出数据

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	…
$u(k)$	-1	-1	-1	-1	1	1	1	-1	1	1	-1	-1	1	-1	1	-1	-1	-1	-1	1	…
$y(k)$	0	-2	-6	-7	-7	-3	5	7	3	-1	5	3	-5	-3	1	-1	1	-5	-7	-7	…

利用这些数据辨识系统的脉冲响应，当 N 取 3 时，有

$$y(k)=\sum_{i=1}^{3} g_i u(k-i) \tag{4.41}$$

令

$$\begin{cases} \boldsymbol{h}(k)=[u(k-1),u(k-2),u(k-3)]^{\mathrm{T}} \\ \boldsymbol{g}=[g_1,g_2,g_3]^{\mathrm{T}} \end{cases} \tag{4.42}$$

若系统的真实脉冲响应记作 $\boldsymbol{g}_0$，则有

$$y(k)=\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{g}_0 \tag{4.43}$$

根据式(4.16)，脉冲响应估计值 $\hat{\boldsymbol{g}}(k)$ 的递推算式为

$$\hat{\boldsymbol{g}}(k+1)=\hat{\boldsymbol{g}}(k)+\boldsymbol{R}(k)\boldsymbol{h}(k)[y(k)-\boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{g}}(k)] \tag{4.44}$$

式中，脉冲响应估计值的初始值取 $\hat{\boldsymbol{g}}=\boldsymbol{0}$，权矩阵 $\boldsymbol{R}(k)$ 取如下形式

$$\boldsymbol{R}^{*}(k)=\frac{1}{\sum_{i=1}^{N}\Lambda_i(k)h_i^2(k)}\mathbf{diag}[\Lambda_1(k),\Lambda_2(k),\Lambda_3(k)] \tag{4.45a}$$

$$\Lambda_1(k)=1,\quad \Lambda_2(k)=\frac{1}{2},\quad \Lambda_3(k)=\frac{1}{4} \tag{4.45b}$$

确定性问题梯度校正递推算法辨识的流程如图 4.3 所示。下面给出 MATLAB 6.0 程序。

```
%梯度校正参数辨识程序，在光盘中的文件名：FLch4GAeg1.m
clear
u=[-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,1,-1,1,-1,-1,-1,-1,1];
y=[0,-2,-6,-7,-7,-3,5,7,3,-1,5,3,-5,-3,1,-1,1,-5,-7,-7];
%画出 u 和 y 图形
figure(1), subplot(2,1,1), stem(u), subplot(2,1,2), stem(y), hold on
k=1:20;plot(k,y)
%给出初始值
h1=[-1,0,0]'; h2=[-1,-1,0]'; g=[0,0,0]'; I=[1,0,0;0,1/2,0;0,0,1/4];
h=[h1,h2,zeros(3,16)];
```

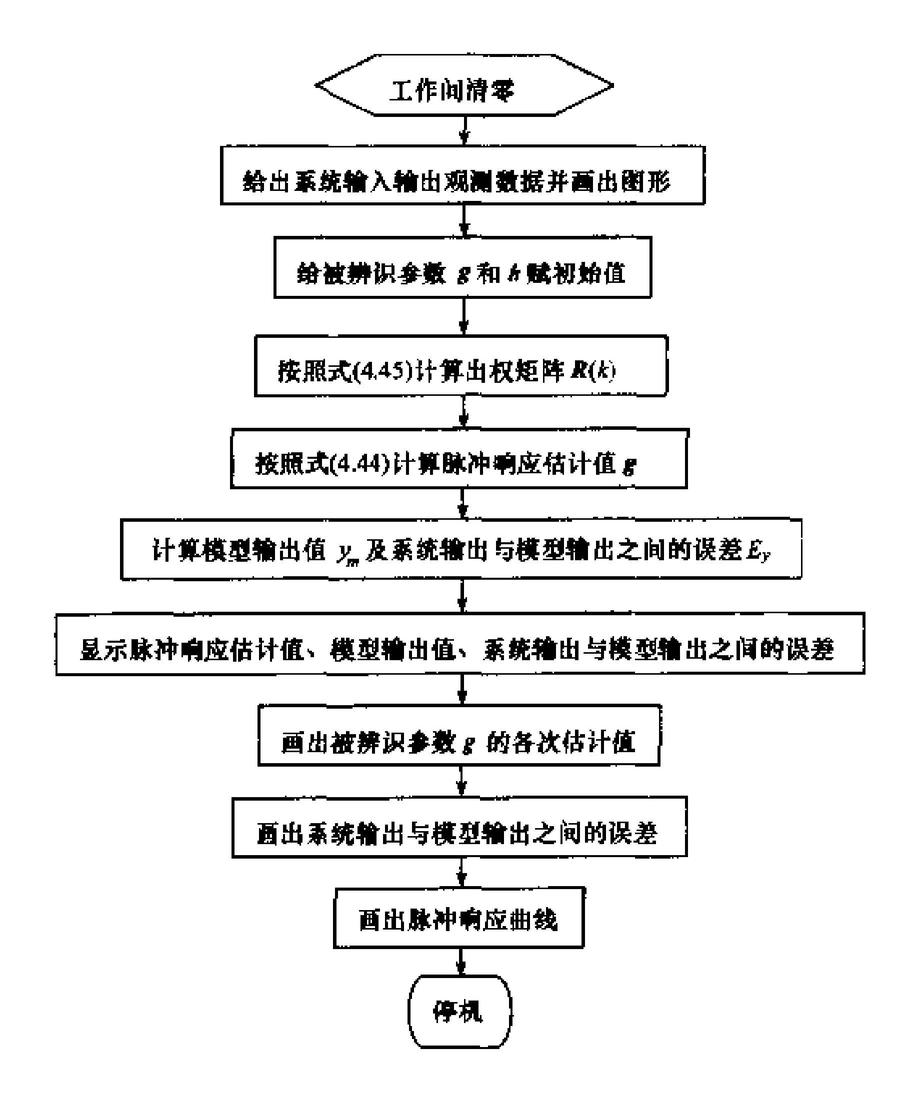


图 4.3　确定性梯度校正辨识的 MATLAB 6.0 程序流程图

```
%计算样本数据 h(k)
for k=3:18
    h(:,k)=[u(k),u(k-1),u(k-2)]';
end
%计算权矩阵 R(k)和 g 的估计值
for k=1:18
    a=h(1,k)^2+(h(2,k)^2)/2+(h(3,k)^2)/4; a1=1/a; R=a1*I; %按照式(4.45a)和(4.45b)计算
%权矩阵
    g(:,k+1)=g(:,k)+R*h(:,k)*(y(k+1)-h(:,k)'*g(:,k)); %按照式(4.44)计算脉冲响应的
%估计值
end
%绘图
g1=g(1,:); g2=g(2,:); g3=g(3,:);
figure(2); k=1:19; subplot(121); plot(k,g1,'r',k,g2,'g',k,g3,'b'), grid on
%计算模型输出 ym 及系统输出与模型输出之间的误差 Ey
for k=1:18
```

```
    ym(k) = h(:,k)' * g(:,k); Ey(k) = y(k+1) - ym(k);
end
k = 1:18; subplot(122); plot(k,Ey), grid on
g, ym, Ey %显示脉冲响应估计值、模型输出及系统输出与模型输出之间的误差
figure(3); x = 0:1:3; y = [0,g(1,18),g(2,18),g(3,18)]; xi = linspace(0,3); yi = interp1(x,y,xi,'cubic');
plot(x,y,'o',xi,yi,'m'), grid on %画出脉冲响应估计值及其三次插值曲线
```

程序执行结果：

g =

$$
\begin{bmatrix}
0 & 2.0000 & 4.6667 & 5.2381 & 5.2381 & 1.5374 & 2.1438 & 2.2393 & 1.9658 & 1.9559 \\
0 & 0 & 1.3333 & 1.6190 & 1.6190 & 3.4694 & 3.7726 & 3.8204 & 3.9571 & 3.9621 \\
0 & 0 & 0 & 0.1429 & 0.1429 & 1.0680 & 0.9164 & 0.9403 & 1.0087 & 1.0062
\end{bmatrix}
$$

$$
\begin{matrix}
2.0063 & 1.9918 & 1.9980 & 1.9953 & 1.9995 & 1.9995 & 1.9995 & 2.0001 & 2.0032 \\
3.9873 & 3.9946 & 3.9977 & 3.9990 & 3.9969 & 3.9969 & 3.9969 & 3.9972 & 3.9988 \\
0.9936 & 0.9972 & 0.9957 & 0.9963 & 0.9974 & 0.9974 & 0.9974 & 0.9972 & 0.9980
\end{matrix}
$$

ym =

[0, - 2.0000, - 6.0000, - 7.0000,3.4762,3.9388,6.8328,2.5213, - 0.9826,4.9118,2.9746,
% - 4.9891, - 2.9953, 1.0073, - 1.0000,1.0000, - 4.9990, - 6.9945]

Ey =

[- 2.0000, - 4.0000, - 1.0000,0.0000, - 6.4762,1.0612,0.1672,0.4787, - 0.0174,0.0882,0.0254,
0.0109, - 0.0047, - 0.0073, - 0.0000,0, - 0.0010, - 0.0055]

图 4.4 表明，递推到第 10 步时，被辨识参数基本上达到了稳定状态，即 $\hat{g}_1 \to 2$，$\hat{g}_2 \to 4$，$\hat{g}_3 \to 1$；此时系统的输出与模型的输出误差也基本达到稳定状态，即 $E_y \to 0$。由于被辨识参数的个数较少，递推校正算法的收敛性比较好，也就是说，输入输出的观测数据量已足够。但从图 4.4 Figure 3 中可看出，由于被辨识参数的个数较少，它还不足以充分显示全部的系统脉冲响应。

为了充分显示出系统的脉冲响应，可以把被辨识参数的个数增加到 5 个。图 4.5 给出了被辨识参数的个数为 5 时的辨识结果。图 4.5 基本显示出了系统的全部脉冲响应过程。但在图 4.5 的辨识中，还是利用上面给出的 20 对输入输出数据。通过图 4.4 与图 4.5 的比较可以看出，在观测数据量一定的情况下，随着被辨识参数的个数的增加，梯度校正辨识算法的收敛性变差，这是由于观测数据不足，递推步数有限造成的。所以，随着被辨识参数的个数的增加，若要保持梯度校正辨识算法的收敛效果，需要同步增加观测数据量。

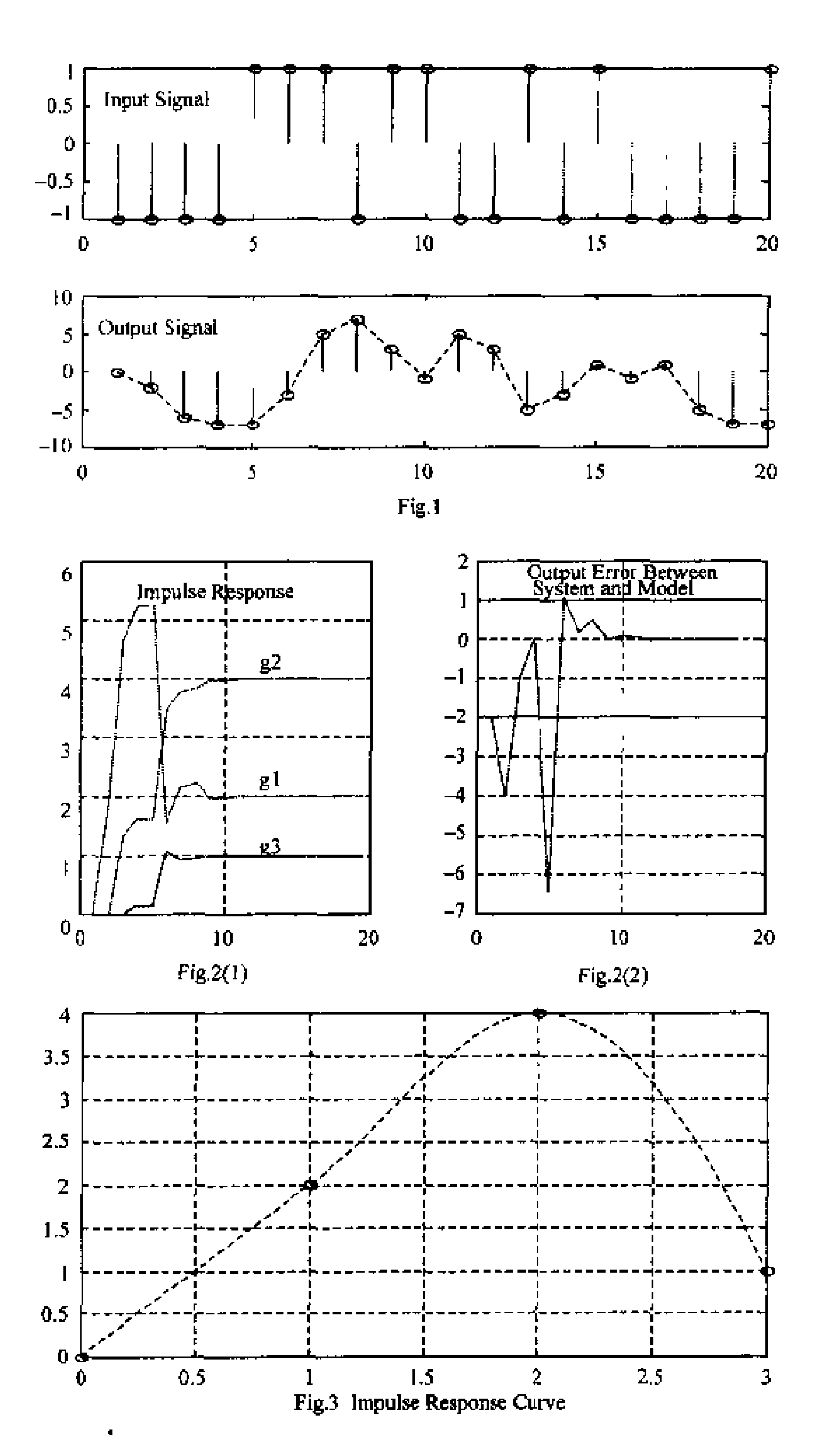


图 4.4　确定性问题梯度校正参数辨识仿真(被辨识参数的个数为 3)

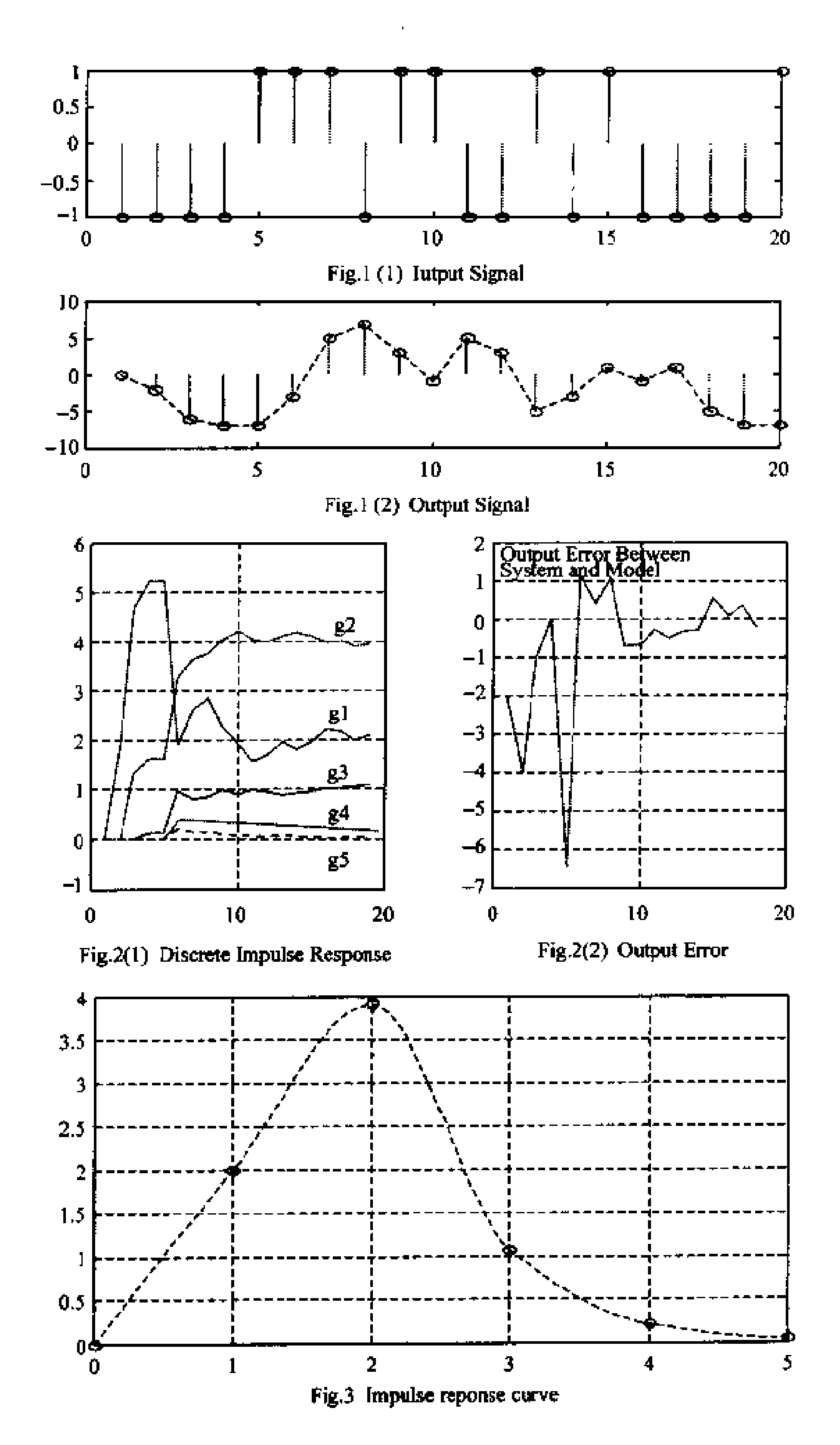


图 4.5 确定性问题梯度校正参数辨识仿真(被辨识参数的个数为 5)

4.3 随机性问题的梯度校正参数辨识方法

4.3.1 随机性问题的提法

确定性问题的梯度校正辨识方法与其他辨识方法相比,最大的优点是计算简单。但是,如果系统的输入输出含有噪声,这种方法就不好用了,为此必须研究随机性问题的梯度校正法。随机性问题的梯度校正法的特点也是计算简单,可用于系统在线实时辨识。但是,它要求事先必须知道噪声的一阶矩和二阶矩的统计特性,这是这种方法的缺陷。

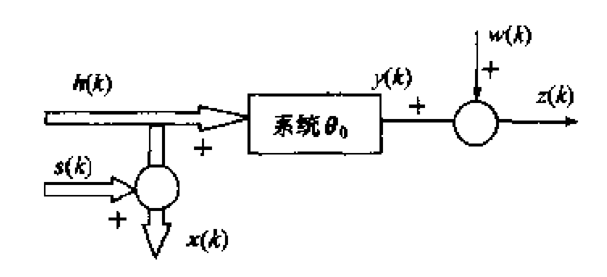


图 4.6 随机性系统

如图 4.6 所示,设系统的输出 $y(k)$是模型参数 $\theta_1,\theta_2,\cdots,\theta_N$ 的线性组合

$$y(k)=h_1(k)\theta_1+h_2(k)\theta_2+\cdots+h_N(k)\theta_N \tag{4.46}$$

输入输出数据均含有测量噪声,即

$$\begin{cases}z(k)=y(k)+w(k)\\ x_i(k)=h_i(k)+s_i(k), \quad i=1,2,\cdots,N\end{cases} \tag{4.47}$$

式中,$w(k)$和 $s_i(k)$为零均值的不相关随机噪声,且

$$E\{s_i(k)s_j(k)\}=\begin{cases}\sigma_{s i}^2, & i=j\\ 0, & i\neq j\end{cases} \tag{4.48}$$

置

$$\begin{cases}\boldsymbol{x}(k)=[x_1(k),x_2(k),\cdots,x_N(k)]^{\mathrm{T}}\\ \boldsymbol{h}(k)=[h_1(k),h_2(k),\cdots,h_N(k)]^{\mathrm{T}}\\ \boldsymbol{s}(k)=[s_1(k),s_2(k),\cdots,s_N(k)]^{\mathrm{T}}\\ \boldsymbol{\theta}=[\theta_1,\theta_2,\cdots,\theta_N]^{\mathrm{T}}\end{cases} \tag{4.49}$$

则

$$\begin{cases}\boldsymbol{x}(k)=\boldsymbol{h}(k)+\boldsymbol{s}(k)\\ z(k)=\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}+w(k)\end{cases} \tag{4.50}$$

现在的问题就是利用输入输出数据 $\boldsymbol{x}(k)$和 $z(k)$来确定参数 $\boldsymbol{\theta}$ 在k 时刻的估计值

$\boldsymbol{\theta}(k)$，使准则函数

$$J(\boldsymbol{\theta})\Big|_{\hat{\boldsymbol{\theta}}(k)} = \frac{1}{2}\varepsilon^2(\boldsymbol{\theta},k)\Big|_{\hat{\boldsymbol{\theta}}(k)} = \min \tag{4.51}$$

式中

$$\varepsilon(\boldsymbol{\theta},k) = z(k) - \boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{\theta} \tag{4.52}$$

下面将讨论这个问题的参数辨识方法。

4.3.2 随机性辨识问题的分类

随机性辨识问题一般分为三类。

1. 第一类随机性辨识问题

图 4.7 所示的随机性系统辨识，如果满足下述条件，则称为第一类随机性辨识问题。

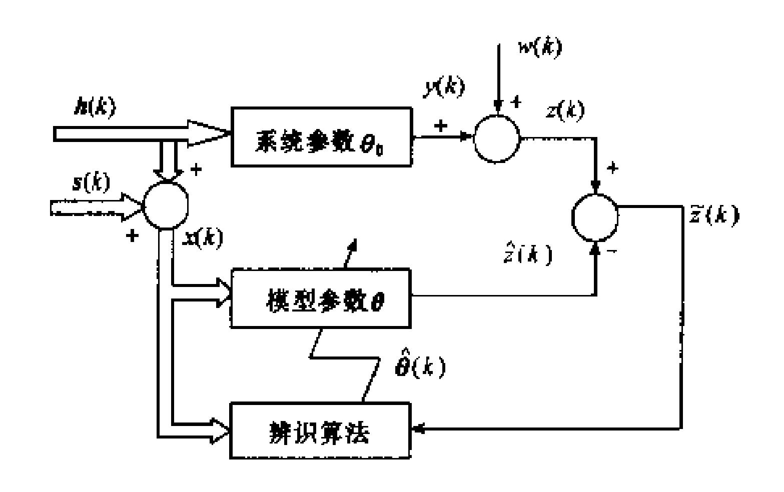


图 4.7　第一类随机性辨识问题

(1) 输入向量 $\boldsymbol{h}(k)$与零均值的测量噪声 $w(k)$是独立统计，即有

$$E\{\boldsymbol{h}(k)w(k)\} = \boldsymbol{0} \tag{4.53}$$

式中，$w(k)$的方差不必先知。

(2) 输入向量 $\boldsymbol{h}(k)$的条件协方差阵是常数阵，且与$\hat{\boldsymbol{\theta}}(k)$无关，即

$$E\{\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k) \mid \hat{\boldsymbol{\theta}}(k)\} = E\{\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)\} = \boldsymbol{\Omega} \tag{4.54}$$

式中，$\boldsymbol{\Omega}$ 是正定的常数阵，但不必先知。

(3) 输入向量的测量噪声 $\boldsymbol{s}(k)$是均值为零，协方差阵为 $\boldsymbol{\Sigma}_s$ 的不相关离散随机向量，且与 $\boldsymbol{h}(k)$和 $w(k)$都是统计独立的，即

$$\begin{cases} E\{\boldsymbol{s}(k)\} = \boldsymbol{0} \\ E\{\boldsymbol{s}(k)\boldsymbol{s}^{\mathrm{T}}(k)\} = \mathbf{diag}\{\sigma_{s_1}^2, \sigma_{s_2}^2, \cdots, \sigma_{s_N}^2\} = \boldsymbol{\Sigma}_s \\ E\{\boldsymbol{s}(k)w(k)\} = \boldsymbol{0} \\ E\{\boldsymbol{h}(k)\boldsymbol{s}^{\mathrm{T}}(k)\} = \boldsymbol{0} \end{cases} \tag{4.55}$$

式中，$\boldsymbol{\Sigma}_s$ 要求先知。

2．第二类随机性辨识问题

如图 4.8 所示随机性系统辨识，如果满足下述条件，则称为第二类随机性辨识问题。

(1) 噪声 $w(k)$可分解成测量噪声 $w_{\mathrm{m}}(k)$和扰动噪声 $w_{\mathrm{d}}(k)$，即

$$w(k) = w_{\mathrm{m}}(k) + w_{\mathrm{d}}(k) \tag{4.56}$$

式中，$w_{\mathrm{d}}(k)$通过“动态环节”与输入向量 $\boldsymbol{h}(k)$相关。已知 $w(k)$的均值为零，但其方差不必先知。

(2) 输入向量 $\boldsymbol{h}(k)$的条件协方差阵是常数阵，且与$\hat{\boldsymbol{\theta}}(k)$无关，即

$$E\{\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k) \mid \hat{\boldsymbol{\theta}}(k)\} = E\{\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)\} = \boldsymbol{\Omega} \tag{4.57}$$

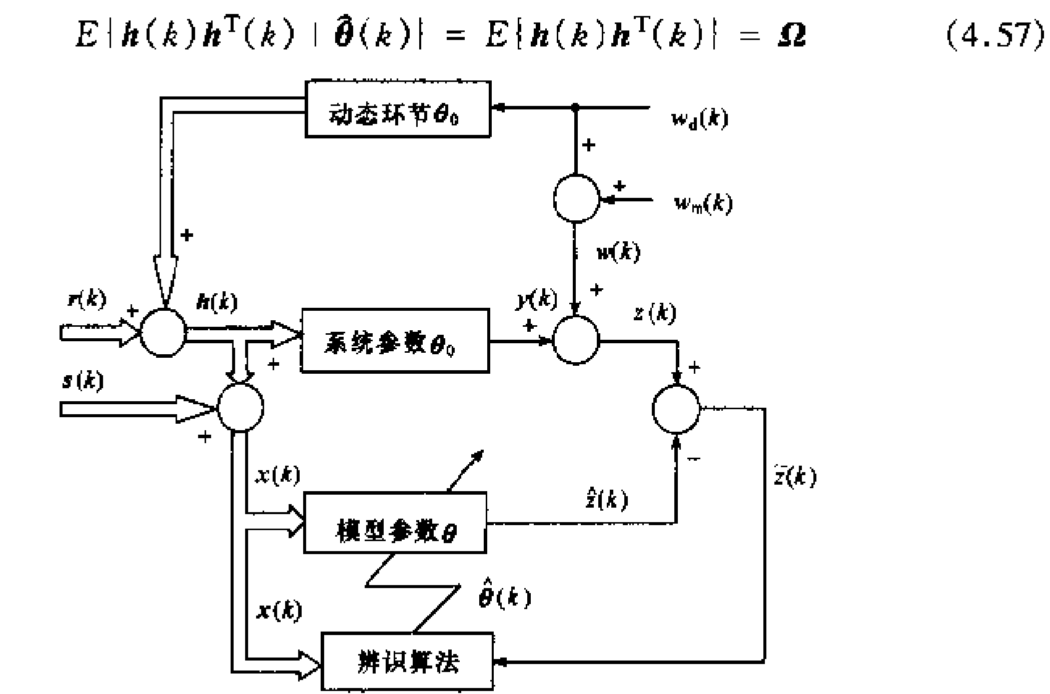


图 4.8 第二类随机性辨识问题

式中，$\boldsymbol{\Omega}$ 是正定常数阵，它也不必先知。

(3) 输入向量的测量噪声 $\boldsymbol{s}(k)$是均值为零，协方差阵为 $\boldsymbol{\Sigma}_s$ 的不相关离散随机向量，且与 $\boldsymbol{h}(k)$和 $w(k)$都是统计独立的，即

$$\begin{cases} E\{\boldsymbol{s}(k)\} = \boldsymbol{0} \\ E\{\boldsymbol{s}(k)\boldsymbol{s}^{\mathrm{T}}(k)\} = \mathbf{diag}\{\sigma_{s_1}^2, \sigma_{s_2}^2, \cdots, \sigma_{s_N}^2\} = \boldsymbol{\Sigma}_s \\ E\{\boldsymbol{s}(k)w(k)\} = \boldsymbol{0} \\ E\{\boldsymbol{h}(k)\boldsymbol{s}^{\mathrm{T}}(k)\} = \boldsymbol{0} \end{cases} \tag{4.58}$$

式中，$\boldsymbol{\Sigma}_s$ 要先知。

第二类随机辨识问题与第一类问题的差别仅在于 $\boldsymbol{h}(k)$与 $w(k)$是否相关。例如，下面的差分方程

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = \omega(k-j) \tag{4.59}$$

式中，$\omega(k)$为零均值的过程噪声。输出 $y(k)$含有测量噪声 $w(k)$，即

$$z(k) = y(k) + w(k) \tag{4.60}$$

置

$$\begin{cases} \boldsymbol{h}(k) = [-y(k-1), \cdots, -y(k-n)]^{\mathrm{T}} \\ \boldsymbol{\theta} = [a_1, a_2, \cdots, a_n]^{\mathrm{T}} \\ e(k) = \omega(k) + w(k) \end{cases} \tag{4.61}$$

则

$$z(k) = \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta} + e(k) \tag{4.62}$$

参数 $\boldsymbol{\theta}$ 的辨识问题是一种典型的随机性辨识问题。当式(4.59)中 $j<2$ 时，它属于第一类辨识问题；当 $j\geqslant 2$ 时，它就变成第二类随机性辨识问题。

3. 第三类随机性辨识问题

第三类随机性辨识问题不仅输入向量 $\boldsymbol{h}(k)$与噪声 $w(k)$相关，而且 $\boldsymbol{h}(k)$与 $\hat{\boldsymbol{\theta}}(k)$也相关，如图 4.9 所示。这类辨识问题难度较大，本节主要讨论第一、二类辨识问题。

4.3.3 随机性问题的梯度校正参数估计方法

随机性问题梯度校正法的基本思想与确定性问题一样，也是利用最速下降原理，从给定的初始值 $\hat{\boldsymbol{\theta}}(0)$出发，沿着准则函数 $J(\boldsymbol{\theta})$的负梯度方向修正估计值 $\hat{\boldsymbol{\theta}}(k)$，直至准则函数 $J(\boldsymbol{\theta})$达到最小值，其数学表达式

$$\hat{\boldsymbol{\theta}}(k+l) = \hat{\boldsymbol{\theta}}(k) - \boldsymbol{R}(k)\,\underset{\boldsymbol{\theta}}{\mathbf{grad}}[J(\boldsymbol{\theta})]\,|_{\boldsymbol{\theta}(k)} \tag{4.63}$$

式中，步长间隔 l 的选择必须使第一、二类随机性辨识问题的条件(2)得到满足；$\boldsymbol{R}(k)$是 N 维的对称矩阵，称作权矩阵；$\underset{\boldsymbol{\theta}}{\mathbf{grad}}[J(\boldsymbol{\theta})]$表示准则函数 $J(\boldsymbol{\theta})$关于 $\boldsymbol{\theta}$ 的梯度。当准则函数 $J(\boldsymbol{\theta})$取式(4.51)时，和确定性问题一样，有

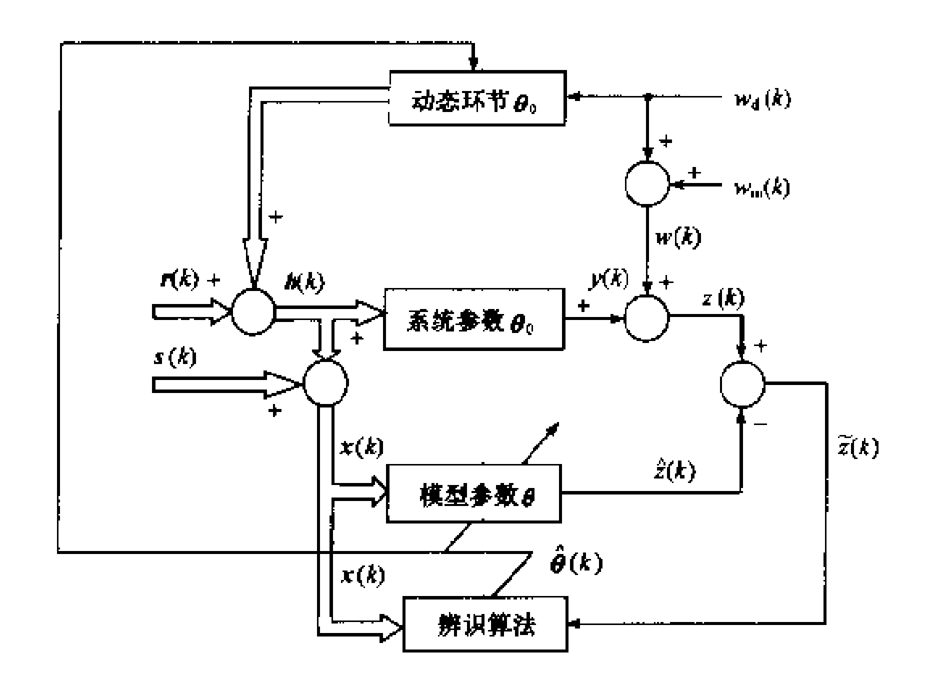


图 4.9 第三类随机性辨识问题

$$\underset{\boldsymbol{\theta}}{\mathbf{grad}}[J(\boldsymbol{\theta})]\,|_{\boldsymbol{\theta}(k)} = -[z(k) - \boldsymbol{x}^{\mathrm{T}}(k)\,\hat{\boldsymbol{\theta}}(k)]\boldsymbol{x}(k) \tag{4.64}$$

则式(4.63)可写成

$$\hat{\boldsymbol{\theta}}(k+1) = \hat{\boldsymbol{\theta}}(k) + \boldsymbol{R}(k)\boldsymbol{x}(k)[z(k) - \boldsymbol{x}^{\mathrm{T}}(k)\,\hat{\boldsymbol{\theta}}(k)] \tag{4.65}$$

此式是随机性问题的梯度校正法的基本公式，它和式(4.57)确定性问题的梯度校正法公式类似。值得注意的是，(4.65)式只能给出参数渐进有偏估计值，即

$$\lim_{k\to\infty} E\{\hat{\boldsymbol{\theta}}(k)\} \neq \boldsymbol{\theta}_0 \tag{4.66}$$

这一事实写成定理 4.2。

定理 4.2 对第二类随机性辨识问题来说，利用式(4.65)获得的参数估计值是渐进有偏的，即

$$\lim_{k\to\infty} E\{\hat{\boldsymbol{\theta}}(k)\} = (\boldsymbol{\Omega} + \boldsymbol{\Sigma}_s)^{-1}(\boldsymbol{T}_2 + \boldsymbol{\Omega}\boldsymbol{\theta}_0) \neq \boldsymbol{\theta}_0 \tag{4.67}$$

式中，$\boldsymbol{\theta}_0$ 表示过程的真实参数，且

$$\begin{cases} \boldsymbol{T}_2 = E\{\boldsymbol{h}(k)w(k)\} \\ \boldsymbol{\Sigma}_s = E\{\boldsymbol{s}(k)\boldsymbol{s}^{\mathrm{T}}(k)\} \\ \boldsymbol{\Omega} = E\{\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)\} \end{cases} \tag{4.68}$$

证明从略。

根据定理 4.2，显然可以得到如下的推论：

推论 4.1 对第一类随机性辨识问题，当输入向量不含测量噪声时，式(4.65)可给出渐进无偏估计值，即有

$$\lim_{k\to\infty} E\{\hat{\boldsymbol{\theta}}(k)\} = \boldsymbol{\theta}_0 \tag{4.69a}$$

且

$$\lim_{k\to\infty} E\{\hat{\boldsymbol{\theta}}(k+l)\} = \lim_{k\to\infty} E\{\hat{\boldsymbol{\theta}}(k)\} + \boldsymbol{R}(k)\boldsymbol{\Omega}\boldsymbol{\theta}_0 - \boldsymbol{R}(k)\boldsymbol{\Omega}\lim_{k\to\infty} E\{\hat{\boldsymbol{\theta}}(k)\} = \boldsymbol{\theta}_0 \tag{4.69b}$$

这时，参数估计值就变成渐进无偏的了，基于这种思想，可以给出随机性辨识问题的渐进无偏估计算法。

1．第一类随机性辨识问题的梯度校正渐进无偏估计算法

对第一类随机性辨识问题来说，利用式(4.65)所获得的参数估计值和第二类问题一样也是渐进有偏的。但根据第一类问题的条件，有 $\boldsymbol{T}_2=\boldsymbol{0}$，故式(4.67)简化为

$$\lim_{k\to\infty} E\{\hat{\boldsymbol{\theta}}(k)\} = (\boldsymbol{\Omega}+\boldsymbol{\Sigma}_s)^{-1}\boldsymbol{\Omega}\boldsymbol{\theta}_0 \tag{4.70}$$

且

$$E\{\hat{\boldsymbol{\theta}}(k+l)\} = E\{\hat{\boldsymbol{\theta}}(k)\} + \boldsymbol{R}(k)\boldsymbol{\Omega}\boldsymbol{\theta}_0 - \boldsymbol{R}(k)(\boldsymbol{\Omega}+\boldsymbol{\Sigma}_s)E\{\hat{\boldsymbol{\theta}}(k)\} \tag{4.71}$$

若在式(4.65)右边增加一项 $\boldsymbol{R}(k)\boldsymbol{\Sigma}_s\hat{\boldsymbol{\theta}}(k)$，则式(4.71)进一步变成

$$E\{\hat{\boldsymbol{\theta}}(k+l)\} = E\{\hat{\boldsymbol{\theta}}(k)\} + \boldsymbol{R}(k)\boldsymbol{\Omega}[\boldsymbol{\theta}_0 - E\{\hat{\boldsymbol{\theta}}(k)\}] \tag{4.72}$$

这时 $\hat{\boldsymbol{\theta}}(k)$ 将成为渐进无偏估计量。

综上分析，第一类随机性辨识问题的梯度校正渐进无偏估计算法应为

$$\boldsymbol{\theta}(k+l) = [\boldsymbol{I}+\boldsymbol{R}(k)\boldsymbol{\Sigma}_s]\hat{\boldsymbol{\theta}}(k) + \boldsymbol{R}(k)\boldsymbol{x}(k)[z(k)-\boldsymbol{x}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k)] \tag{4.73}$$

式中，步长间隔 l 的选择必须使第一类问题的条件(2)得到满足；$\boldsymbol{R}(k)$ 为权矩阵，稍后将进一步讨论 l 和 $\boldsymbol{R}(k)$ 的选择问题；$\boldsymbol{\Sigma}_s$ 是输入向量 $\boldsymbol{h}(k)$ 的测量噪声 $\boldsymbol{s}(k)$ 的协方差阵，需先知道；$\boldsymbol{x}(k)$ 和 $z(k)$ 是可测的输入输出数据。

2．第二类随机性辨识问题的梯度校正渐进无偏估计算法

在第二类随机性辨识问题中，由于 $\boldsymbol{T}_2=E\{\boldsymbol{h}(k)w(k)\}\neq\boldsymbol{0}$，所以为了获得参数的渐进无偏估计，类似于第一类问题的分析，必须在式(4.65)的右边增加两项，$\boldsymbol{R}(k)\boldsymbol{\Sigma}_s\hat{\boldsymbol{\theta}}(k)$ 和 $-\boldsymbol{R}(k)\boldsymbol{T}_2$。于是第二类随机性辨识问题的梯度校正渐进无偏估计写成

$$\hat{\boldsymbol{\theta}}(k+l) = [\boldsymbol{I}+\boldsymbol{R}(k)\boldsymbol{\Sigma}_s]\hat{\boldsymbol{\theta}}(k) - \boldsymbol{R}(k)\boldsymbol{T}_2 + \boldsymbol{R}(k)\boldsymbol{x}(k)[z(k)-\boldsymbol{x}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k)] \tag{4.74}$$

式中,变量的含义与式(4.73)相同。这里不仅要求 $\boldsymbol{\Sigma}_s$ 必须是已知的,而且 T_2 也是已知的。从图 4.8 可以看出,$\boldsymbol{h}(k)$是参数 $\boldsymbol{\theta}_0$ 的函数,也就是说 $\boldsymbol{T}_2$ 是 $\boldsymbol{\theta}_0$ 的函数。不妨设 $\boldsymbol{T}_2$ 是 $\boldsymbol{\theta}_0$ 的线性函数,写成

$$\boldsymbol{T}_2 = \boldsymbol{\lambda} + \boldsymbol{M}\boldsymbol{\theta}_0 \tag{4.75}$$

式中,$\boldsymbol{\lambda}$ 是 N 维向量,$\boldsymbol{M}$ 是 $N\times N$ 维矩阵,在特定条件下它们可以事先确定。当然,$\boldsymbol{T}_2$ 与 $\boldsymbol{\theta}_0$ 的关系不一定都写成式(4.75)的形式,也可以根据实际问题具体确定函数关系。

下面着重讨论 $\boldsymbol{T}_2$ 与 $\boldsymbol{\theta}_0$ 的关系取式(4.75)时,第二类随机性辨识问题的梯度校正无偏算法。

根据式(4.75),$\boldsymbol{T}_2$ 的估计值应为

$$\hat{\boldsymbol{T}}_2 = \boldsymbol{\lambda} + \boldsymbol{M}\hat{\boldsymbol{\theta}}(k) \tag{4.76}$$

若式(4.74)中的 $\boldsymbol{T}_2$ 用$\hat{\boldsymbol{T}}_2$ 代替,则第二类随机性辨识问题的梯度校正估计算法可写成

$$\begin{aligned}\hat{\boldsymbol{\theta}}(k+l) = [\boldsymbol{I} + \boldsymbol{R}(k)\boldsymbol{\Sigma}_s - \boldsymbol{R}(k)\boldsymbol{M}]\,\hat{\boldsymbol{\theta}}(k) - \boldsymbol{R}(k)\boldsymbol{\lambda}\\ + \boldsymbol{R}(k)\boldsymbol{x}(k)[z(k) - \boldsymbol{x}^{\mathrm{T}}(k)\,\hat{\boldsymbol{\theta}}(k)]\end{aligned} \tag{4.77}$$

类似于定理 4.2,有

$$E\{\hat{\boldsymbol{\theta}}(k+l)\} = E\{\hat{\boldsymbol{\theta}}(k)\} + \boldsymbol{R}(k)(\boldsymbol{\Omega} + \boldsymbol{M})\boldsymbol{\theta}_0 - \boldsymbol{R}(k)(\boldsymbol{\Omega} + \boldsymbol{M})E\{\hat{\boldsymbol{\theta}}(k)\} \tag{4.78}$$

当 $k\to\infty$时,可得

$$\lim_{k\to\infty} E\{\hat{\boldsymbol{\theta}}(k)\} = \boldsymbol{\theta}_0 \tag{4.79}$$

可见,算法(4.77)给出了渐进无偏估计。其中 l 的选择必须保证第二类辨识问题的条件(2)成立;$\boldsymbol{R}(k)$为权矩阵,它的具体选择下面将具体讨论;$\boldsymbol{\lambda}$ 和 $\boldsymbol{M}$ 为参变量,结合实际问题可以惟一确定;其他变量与第一类辨识问题一样。

3. 步长间隔的选择

不论第一类随机性辨识问题的渐进无偏估计算法(4.73),还是第二类随机性辨识问题的渐进无偏估计算法(4.77),都要选择步长间隔 l 使输入向量 $\boldsymbol{h}(k)$与参数估计值$\hat{\boldsymbol{\theta}}(k)$不相关。式(4.73)或(4.77)表明,$\boldsymbol{\theta}(k+l)$是$\hat{\boldsymbol{\theta}}(k)$、$\boldsymbol{h}(k)$、$\boldsymbol{s}(k)$和 $w(k)$的函数,可表示成

$$\hat{\boldsymbol{\theta}}(k+l) = f(\hat{\boldsymbol{\theta}}(k), \boldsymbol{h}(k), \boldsymbol{s}(k), w(k)) \tag{4.80}$$

当 $k=0$ 时,有

$$\hat{\boldsymbol{\theta}}(k) = f(\hat{\boldsymbol{\theta}}(0), \boldsymbol{h}(0), \boldsymbol{s}(0), w(0)) = g_1(\hat{\boldsymbol{\theta}}(0), \boldsymbol{h}(0), \boldsymbol{s}(0), w(0)) \tag{4.81}$$

当 $k=l$ 时,有

$$\begin{aligned}\hat{\boldsymbol{\theta}}(2l) &= f(\hat{\boldsymbol{\theta}}(l), \boldsymbol{h}(l), \boldsymbol{s}(l), w(l)) \\ &= f(f(\hat{\boldsymbol{\theta}}(0)), \boldsymbol{h}(0), \boldsymbol{s}(0), w(0), \boldsymbol{h}(l), \boldsymbol{s}(l), w(l)) \\ &= g_2(\hat{\boldsymbol{\theta}}(0), \boldsymbol{h}(0), \boldsymbol{s}(0), w(0), \boldsymbol{h}(l), \boldsymbol{s}(l), w(l))\end{aligned} \tag{4.82}$$

以此类推,有

$$\begin{aligned}\hat{\boldsymbol{\theta}}(ml) &= f(\hat{\boldsymbol{\theta}}(ml-l), \boldsymbol{h}(ml-l), \boldsymbol{s}(ml-l), w(ml-l)) \\ &= g_m(\hat{\boldsymbol{\theta}}(0), \boldsymbol{h}(0), \boldsymbol{s}(0), w(0), \cdots, \hat{\boldsymbol{\theta}}(ml-l), \boldsymbol{h}(ml-l), \\ &\quad \boldsymbol{s}(ml-l), w(ml-l))\end{aligned} \tag{4.83}$$

式中, $f(\cdot)$、$g_1(\cdot)$、$g_2(\cdot)$、…、$g_m(\cdot)$表示某种函数关系。式(4.83)表明。$k=ml$ 时刻的参数估计值 $\hat{\boldsymbol{\theta}}(k)$与 $k-l$ 时刻以前的信息(包括输入向量 $h(0),\cdots,h(k-l)$,输入测量噪声 $s(0),\cdots,s(k-l)$和输出测量噪声 $w(0),\cdots,w(k-l)$)是相关的。这样,选择 l 使输入向量$h(k)$与参数估计值$\hat{\boldsymbol{\theta}}(k)$不相关问题就转变成选择 l 使$h(k)$与$(k-l)$时刻以前的信息(包括 $h(i)$、$s(i)$和 $w(i)$, $i=0,1,2,\cdots,k-l$)不相关问题。根据第一、二类随机性辨识问题的条件,已知 $h(k)$与$(k-l)$时刻以前的 $s(i)$和 $w(i)$, $i=0,1,2,\cdots,k-l$ 不相关,所以只要选择 l 使$h(k)$与 $h(k-l)$不相关,就能使第一、二类随机性辨识问题的条件(2)成立,保证式(4.73)和(4.77)都是渐进无偏估计算法。也就是说,步长间隔 l 的选择必须使输入向量$h(k)$与 $h(k-l)$是统计不相关的。这是选择 l 的基本出发点,具体问题还要具体考虑。比如,如果系统的模型采用 n 阶差分方程形式,则步长间隔 l 不能小于阶次 n,以保证 $h(k)$与 $h(k-l)$不相关。

4. 权矩阵 $\boldsymbol{R}(k)$的选择

第一、二类随机性辨识问题的梯度校正算法(4.73)和(4.77),当选择 l 使$\boldsymbol{h}(k)$与 $\boldsymbol{h}(k-l)$不相关时,它们都是渐进无偏估计算法。但是,渐进无偏估计不一定是均方一致估计或依概率 1 一致估计,即有

$$\lim_{k\to\infty} E\{\hat{\boldsymbol{\theta}}(k)\} = \boldsymbol{\theta}_0 \tag{4.84}$$

但不一定有

$$\lim_{k\to\infty} E\{\|\hat{\boldsymbol{\theta}}(k) - \boldsymbol{\theta}_0\|^2\} = 0 \tag{4.85}$$

或概率为 1,即

$$\text{Prob}\{\lim_{k\to\infty}\hat{\boldsymbol{\theta}}(k)=\boldsymbol{\theta}_0\}=1 \tag{4.86}$$

为了使式(4.73)和(4.77)算法同时又是一致估计,需要适当选择权矩阵 $\boldsymbol{R}(k)$。下面将随机性问题的梯度校正参数辨识算法及权矩阵的选择归纳成定理 4.3。

定理 4.3　第一类随机辨识问题的梯度校正渐进无偏估计算法为

$$\boldsymbol{\theta}(k+l)=[\boldsymbol{I}+\boldsymbol{R}(k)\boldsymbol{\Sigma}_s]\hat{\boldsymbol{\theta}}(k)+\boldsymbol{R}(k)\boldsymbol{x}(k)\varepsilon(k) \tag{4.87}$$

第二类随机性辨识问题的梯度校正渐进无偏估计算法为

$$\hat{\boldsymbol{\theta}}(k+l)=[\boldsymbol{I}+\boldsymbol{R}(k)\boldsymbol{\Sigma}_s-\boldsymbol{R}(k)\boldsymbol{M}]\hat{\boldsymbol{\theta}}(k)-\boldsymbol{R}(k)\boldsymbol{\lambda}+\boldsymbol{R}(k)\boldsymbol{x}(k)\varepsilon(k) \tag{4.88}$$

式中,残差 $\varepsilon(k)=z(k)-\boldsymbol{x}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k)$;$\boldsymbol{x}(k)$与 $z(k)$是可测输入输出数据;$\boldsymbol{\Sigma}_s$ 是输入向量$\boldsymbol{h}(k)$测量噪声的协方差阵;参变量 $\boldsymbol{\lambda}$ 和$\boldsymbol{M}$ 根据具体问题事先确定;步长间隔 l 必须使输入向量$\boldsymbol{h}(k)$与 $\boldsymbol{h}(k-l)$不相关;且有

$$\begin{cases}E\{w^2(k)\}<\infty\\ E\{\|\boldsymbol{x}(k)\|^2\boldsymbol{x}(k)\boldsymbol{x}^{\mathrm{T}}(k)\mid\hat{\boldsymbol{\theta}}(k)\}<\infty\\ E\{\|\boldsymbol{x}(k)\|^2w^2(k)\mid\hat{\boldsymbol{\theta}}(k)\}<\infty\end{cases} \tag{4.89}$$

$w(k)$是输出的测量噪声。如果权矩阵选择如下形式

$$\boldsymbol{R}(k)=c(k)\mathbf{diag}[\Lambda_1(k),\Lambda_2(k),\cdots,\Lambda_N(k)] \tag{4.90}$$

式中

$$\begin{cases}0<\Lambda_L\leqslant\Lambda_i(k)\leqslant\Lambda_H<\infty,i=1,2,\cdots,N\ (N=\dim\boldsymbol{\theta})\\ c(k)>0,\forall k;\lim\limits_{k\to\infty}c(k)=0\\ \sum\limits_{k=1}^{\infty}c(k)\to\infty;\sum\limits_{k=1}^{\infty}c^2(k)<\infty\end{cases} \tag{4.91}$$

则参数估计值$\hat{\boldsymbol{\theta}}(k+l)$在均方意义下是一致收敛的,或是依概率 1 一致收敛的。

定理 4.3 的几点说明:

(1) 条件(4.89)是很弱的,一般问题都能解决。

(2) 权矩阵 $\boldsymbol{R}(k)$中 $c(k)$可取

$$c(k)=\frac{1}{k^p},\quad\frac{1}{2}<p\leqslant1;k>0 \tag{4.92}$$

它能满足式(4.91)的要求。

(3) 式(4.91)是保证$\hat{\boldsymbol{\theta}}(k)$一致收敛条件,但$\hat{\boldsymbol{\theta}}(k)$不一定有最快的收敛速度。为了加快$\hat{\boldsymbol{\theta}}(k)$的收敛速度,权矩阵 $\boldsymbol{R}(k)$可分段选择。在递推计算前期,$\boldsymbol{R}(k)$可用 Lyapunov 最佳权值矩阵形式,如式(4.39)所示,以便加速$\hat{\boldsymbol{\theta}}(k)$的收敛速度。在

递推计算后期，$\boldsymbol{R}(k)$必须采用式(4.90)的形式，以保证$\hat{\boldsymbol{\theta}}(k)$的一致收敛的。

例 4.1 随机性梯度校正法在三阶空气动力学系统中的应用[83]。

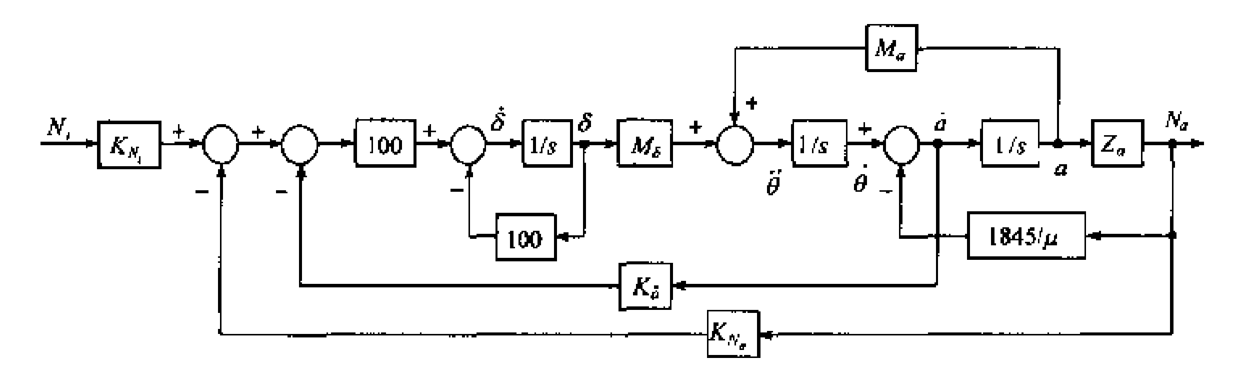


图 4.10 三阶空气动力学系统

设一个三阶的空气动力学系统如图 4.10 所示。图中，N 是沿着 z 轴的加速度输入量；δ 是表面倾斜量；M_δ 是表面倾斜量系数；$\ddot{\theta}$ 是刚体加速度；a 是仰角；M_a 是转矩系数；μ 是轴向速度；N_a 是沿着 z 轴的加速度输出量；Z_a 是加速度系数；K_{N_i}、$K_{\dot{a}}$、K_{N_a} 为对应控制器的增益；M_a、M_δ 和 Z_a 是时变参数。由图 4.10 直接可得

$$\begin{cases}\ddot{\theta}(k) = M_a a(k) + M_\delta(k) \\ N_a(k) = Z_a a(k)\end{cases} \tag{4.93}$$

式中，$\ddot{\theta}(k)$、$a(k)$、$\delta(k)$、$N_a(k)$是可测的变量，其测量噪声分别是均值为零、方差为 $\sigma_{\ddot{\theta}}^2$、σ_a^2、σ_δ^2、σ_N^2 的不相关随机噪声，记作 $w_{\ddot{\theta}}$、w_a、w_δ、w_N，对应的测量值为

$$\ddot{\theta}_m(k) = \ddot{\theta}(k) + w_{\ddot{\theta}}, \qquad a_m(k) = a(k) + w_a \tag{4.94a}$$

$$\delta_m(k) = \delta(k) + w_\delta, \qquad N_{a_m}(k) = N_a(k) + w_N \tag{4.94b}$$

如果控制器的增益 K_{N_i}、$K_{\dot{a}}$、K_{N_a} 满足如下关系

$$\begin{cases}K_{N_i} = \dfrac{c_2}{100 M_\delta Z_a} \\ K_{\dot{a}} = \dfrac{c_1 - 100\left(\dfrac{1845 Z_a}{\mu}\right) + M_a}{100 M_\delta} \\ K_{N_a} = \dfrac{c_2 + 100 M_a}{100 M_\delta - Z_a}\end{cases} \tag{4.95}$$

根据梅逊公式，有

$$\frac{N_a(s)}{N_i(s)} = \frac{c_2}{s^3 + \left(100 + \frac{1845Z_a}{\mu}\right)s^2 + c_1 s + c_2} \tag{4.96}$$

设 $c_1 = 1400, c_2 = 14000$,且$\frac{1845Z_a}{\mu}$很小,那么

$$\frac{N_a(s)}{N_i(s)} = \frac{14000}{s^3 + 100s^2 + 1400s + 14000} \tag{4.97}$$

这就是说,当控制器增益 K_{N_i}、$K_{\dot{a}}$、K_{N_a}满足式(4.95)时,系统的闭环传递函数为(4.97),其闭环阶跃响应的带宽为2cps.,衰减率为0.6,系统无静态偏差。为了使系统始终具有这样的闭环阶跃响应特性,必须使控制器增益 K_{N_i}、$K_{\dot{a}}$、K_{N_a} 随 M_a、M_δ 和 Z_a 变化,实现自适应控制。但是,M_a、M_δ 和 Z_a 是未知的时变参数,这里用梯度校正法获得它们的实时估计值,并代入式(4.95),使增益 K_{N_i}、$K_{\dot{a}}$、K_{N_a} 适应系统参数的变化。

根据所给的条件,参数 M_a、M_δ 和 Z_a 的辨识问题属于第一类随机性辨识问题。根据定理4.3,参数 M_a、M_δ 和 Z_a 的梯度校正渐近无偏估计算法为

(1) 当 $k \leqslant k_1$ 时(k_1 为某选定的整数)

$$\hat{M}_a(k+1) = \left[1 + \frac{\Lambda_1\sigma_a^2}{\Lambda_1 a_m^2(k) + \Lambda_2\delta_m^2(k)}\right]\hat{M}_a(k) + \frac{\Lambda_1 a_m(k)\tilde{\ddot{\theta}}_m(k)}{\Lambda_1 a_m^2(k) + \Lambda_2\delta_m^2(k)} \tag{4.98a}$$

$$\hat{M}_\delta(k+1) = \left[1 + \frac{\Lambda_2\sigma_\delta^2}{\Lambda_1 a_m^2(k) + \Lambda_2\delta_m^2(k)}\right]\hat{M}_\delta(k) + \frac{\Lambda_2 \delta_m(k)\tilde{\ddot{\theta}}_m(k)}{\Lambda_1 a_m^2(k) + \Lambda_2\delta_m^2(k)} \tag{4.98b}$$

$$\hat{Z}_a(k+1) = \left[1 + \frac{\sigma_N^2}{\sigma_m^2(k)}\right]\hat{Z}_a(k) + \frac{\tilde{N}_a(k)}{a_m(k)} \tag{4.98c}$$

式中,Λ_1 和 Λ_2 为加权值,需要事先确定;σ_a^2、σ_δ^2 和σ_N^2 为噪声方差,假设是已知的;且

$$\begin{cases} \tilde{\ddot{\theta}}_m(k) = \ddot{\theta}_m(k) - \hat{M}_a(k)a_m(k) - \hat{M}_\delta(k)\delta_m(k) \\ \tilde{N}_a(k) = N_{a_m}(k) - \hat{Z}_a(k)a_m(k) \end{cases} \tag{4.99}$$

(2) 当 $k > k_1$ 时

$$
\begin{cases}
\hat{M}_a(k+1) = \left(1 + \dfrac{\Lambda_1 \sigma_a^2}{k}\right)\hat{M}_a(k) + \dfrac{\Lambda_1 a_m(k)}{k}\tilde{\ddot{\theta}}(k) \\
\hat{M}_\delta(k+1) = \left(1 + \dfrac{\Lambda_2 \sigma_\delta^2}{k}\right)\hat{M}_\delta(k) + \dfrac{\Lambda_2 \delta_m(k)}{k}\tilde{\ddot{\theta}}(k) \\
\hat{Z}_\delta(k+1) = \left(1 + \dfrac{\sigma_N^2}{k}\right)\hat{Z}_\delta(k) + \dfrac{a_m(k)}{k}\tilde{N}_a(k)
\end{cases}
\tag{4.100}
$$

递推计算的前期($k \leqslant k_1$),采用算式(4.98),目的为了加快参数估计值的收敛速度;递推计算的后期($k > k_1$),采用算式(4.100),可保证参数估计值是一致收敛的。

下面给出该例的模拟结果。已知 $\sigma_a^2 = 0.061, \sigma_\delta^2 = 0.016, \sigma_\theta^2 = 3.0$; $M_a = 500$, $M_\delta = 1000$;取 $\hat{M}_a(0) = 250, \hat{M}_\delta(0) = 500, \Lambda_1 = 0.1, \Lambda_2 = 1$; $k_1 = 52$;加速度输入量 N_i 如图 4.11 所示,作用时间为 0.5s;$\ddot{\theta}_m(k)$、$a_m(k)$ 和 $\delta_m(k)$ 的采样时间为 0.01s;采用算式(4.98)~(4.100),递推计算 52 步后,参数 M_a 和 M_δ 的估计值就基本上稳定在 500 和 1000 附近。利用参数辨识的结果,代入式(4.95),计算控制器的增益,组成自适应控制系统,可使系统的闭环阶跃响应特性始终满足设计要求。

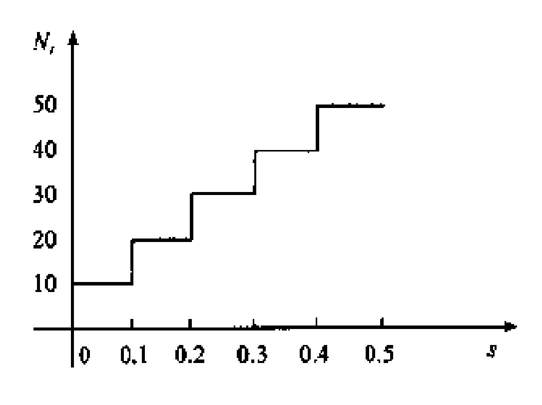


图 4.11　加速度输入量 N_i

该例说明,在一定的条件下,梯度校正法可用于实时在线辨识。它的显著优点是计算最小,但它要求预先知道噪声的统计特性,这是它的缺点。

4.4　梯度校正法在动态过程辨识中的应用

本节将研究如何利用梯度校正法来解决状态方程和差分方程两种模型的参数辨识问题,其中如何确定参变量 $\boldsymbol{\lambda}$ 和 $\boldsymbol{M}$ 是问题的关键。

4.4.1 状态方程的参数辨识

设一个 SISO 系统用状态方程描述如下：

$$\begin{cases} \bar{\boldsymbol{x}}(k+1) = \bar{\boldsymbol{A}}\bar{\boldsymbol{x}}(k) + \bar{\boldsymbol{b}}u(k) + \bar{\boldsymbol{d}}\omega(k) \\ y(k) = \bar{\boldsymbol{c}}\bar{\boldsymbol{x}}(k) \end{cases} \tag{4.101}$$

式中

$$\bar{\boldsymbol{A}} = \left[\begin{array}{c:c} \boldsymbol{0} & \boldsymbol{I}_{n-1} \\ \hdashline \multicolumn{2}{c}{-a_n, -a_{n-1}, \cdots, -a_1} \end{array}\right] \tag{4.102a}$$

$$\begin{cases} \bar{\boldsymbol{b}} = [\bar{b}_1, \bar{b}_2, \cdots, \bar{b}_n]^{\mathrm{T}} \\ \bar{\boldsymbol{d}} = [\bar{d}_1, \bar{d}_2, \cdots, \bar{d}_n]^{\mathrm{T}} \\ \bar{\boldsymbol{c}} = [1, 0, \cdots, 0] \end{cases} \tag{4.102b}$$

$\omega(k)$是均值为零、方差为 σ_ω^2 的不相关随机系统噪声；$\bar{\boldsymbol{d}}$ 是噪声模型参数向量，假定它是已知的；$\bar{\boldsymbol{A}}$ 和 $\bar{\boldsymbol{b}}$ 为未知待辨识的模型参数；输入输出变量 $u(k)$和 $y(k)$对应的测量值记作

$$\begin{cases} x(k) = u(k) + s(k) \\ z(k) = y(k) + \omega(k) \end{cases} \tag{4.103}$$

$s(k)$和 $\omega(k)$分别是均值为零、方差为 σ_s^2 和σ_ω^2 的不相关随机测量噪声，且 $s(k)$和 $\omega(k)$是统计独立的。

状态方程(4.101)对应的差分方程为

$$\begin{aligned} & y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) \\ = \; & b_1 u(k-1) + \cdots + b_n u(k-n) + d_1 \omega(k-1) + \cdots + d_n \omega(k-n) \end{aligned} \tag{4.104}$$

式中

$$\begin{cases} \boldsymbol{b} = [b_1, b_2, \cdots, b_n]^{\mathrm{T}} = \boldsymbol{P}\bar{\boldsymbol{b}} \\ \boldsymbol{d} = [d_1, d_2, \cdots, d_n]^{\mathrm{T}} == \boldsymbol{P}\bar{\boldsymbol{d}} \\ \boldsymbol{p} = \begin{bmatrix} 1 & & & & 0 \\ a_1 & 1 & & & \\ a_2 & a_1 & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \\ a_{n-1} & a_{n-2} & a_2 & a_1 & 1 \end{bmatrix} \end{cases} \tag{4.105}$$

置

$$\begin{cases} \boldsymbol{h}(k) = [y(k-1),\cdots,y(k-n),u(k-1),\cdots,u(k-n)]^{\mathrm{T}} \\ \boldsymbol{x}(k) = [z(k-1),\cdots,z(k-n),x(k-1),\cdots,x(k-n)]^{\mathrm{T}} \\ \boldsymbol{\theta} = [-a_1,\cdots,-a_n,b_1,\cdots,b_n]^{\mathrm{T}} \\ \boldsymbol{s}(k) = [\omega(k-1),\cdots,\omega(k-n),s(k-1),\cdots,s(k-n)]^{\mathrm{T}} \\ e(k) = \omega(k) + d_1\omega(k-1) + \cdots + d_n\omega(k-n) = \omega(k) + \boldsymbol{\omega}^{\mathrm{T}}(k)\boldsymbol{d} \\ \boldsymbol{\omega}(k) = [\omega(k-1),\cdots,\omega(k-n)]^{\mathrm{T}} \end{cases} \tag{4.106}$$

则

$$\begin{cases} \boldsymbol{x}(k) = \boldsymbol{h}(k) + \boldsymbol{s}(k) \\ z(k) = \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta} + e(k) \end{cases} \tag{4.107}$$

与式(4.50)比较可知,状态方程(4.101)的参数辨识问题已化成第二类随机性梯度校正参数辨识问题。应用定理4.3,可得参数 $\boldsymbol{\theta}$ 的渐进无偏估计算法为

$$\hat{\boldsymbol{\theta}}(k+l) = [\boldsymbol{I} + \boldsymbol{R}(k)\boldsymbol{\Sigma}_s - \boldsymbol{R}(k)\boldsymbol{M}]\,\hat{\boldsymbol{\theta}}(k) - \boldsymbol{R}(k)\boldsymbol{\lambda} + \boldsymbol{R}(k)\boldsymbol{x}(k)\varepsilon(k) \tag{4.108}$$

式中,残差 $\varepsilon(k) = z(k) - \boldsymbol{x}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k)$;$\boldsymbol{x}(k)$由可测的输入输出数据 $x(\cdot)$和 $z(\cdot)$组成;步长间隔 l 取大于n 的值,以保证 $\boldsymbol{h}(k)$与 $\boldsymbol{h}(k-l)$不相关;$\boldsymbol{\Sigma}_s$ 表示噪声向量$\boldsymbol{s}(k)$的协方差阵,根据 $\boldsymbol{s}(k)$的定义,有

$$\boldsymbol{\Sigma}_s = E\{\boldsymbol{s}(k)\boldsymbol{s}^{\mathrm{T}}(k)\} = \left[\begin{array}{c|c} \sigma_\omega^2\boldsymbol{I}_n & \boldsymbol{0} \\ \hline \boldsymbol{0} & \sigma_s^2\boldsymbol{I}_n \end{array}\right] \tag{4.109}$$

要求 $\boldsymbol{\Sigma}_s$ 是已知的;权矩阵 $\boldsymbol{R}(k)$取

$$\begin{cases} \boldsymbol{R}(k) = \dfrac{1}{k^p}\mathbf{diag}[\Lambda_1(k),\Lambda_2(k),\cdots,\Lambda_{2n}(k)] \\ \dfrac{1}{2} < p \leqslant 1;0 < \Lambda_L \leqslant \Lambda_i(k) \leqslant \Lambda_H < \infty, i = 1,2,\cdots,2n \end{cases} \tag{4.110}$$

参变量 $\boldsymbol{\lambda}$ 和$\boldsymbol{M}$ 满足下列关系

$$E\{\boldsymbol{h}(k)e(k)\} = \boldsymbol{\lambda} + \boldsymbol{M}\boldsymbol{\theta}_0 \tag{4.111}$$

$\boldsymbol{\theta}_0$ 为系统参数真值。

综上所述,若确定了参变量 $\boldsymbol{\lambda}$ 和$\boldsymbol{M}$,则参数 $\boldsymbol{\theta}$ 的估计值可按式(4.108)递推计算。下面具体讨论参变量 $\boldsymbol{\lambda}$ 和$\boldsymbol{M}$ 的求法。值得注意的是:对不同的问题,$\boldsymbol{\lambda}$ 和$\boldsymbol{M}$ 的求法是不同的。

(1) 解状态方程(4.101),得

$$\bar{x}(k) = \bar{A}^k\bar{x}(0) + \sum_{i=1}^{k} \bar{A}^{k-i}[\bar{b}u(i-1) + \bar{d}\omega(i-1)] \tag{4.112}$$

且输出变量可表示成

$$y(k) = \bar{c}\bar{A}^k\bar{x}(0) + \sum_{i=1}^{k} \bar{c}\bar{A}^{k-i}[\bar{b}u(i-1) + \bar{d}\omega(i-1)] \tag{4.113}$$

(2) 确定 $E\{h(k)s(k)\}$与 θ_0 的函数关系。根据式(4.106)中 $h(k)$、$e(k)$的定义，利用式(4.113)的结果，同时注意到噪声 $\omega(k)$和 $w(k)$的统计特性分别是零均值的白噪声，且相互独立，并与 $u(k)$不相关，则有

$$\begin{aligned}
&E\{y(k-1)e(k)\}\\
&= \sum_{i=1}^{k-1} \bar{c}\bar{A}^{k-1-i}\bar{d}E\{\omega(i-1)\boldsymbol{\omega}^{\mathrm{T}}(k)\}\boldsymbol{d} = \sigma_\omega^2[0,\bar{d}_1,\bar{d}_2,\cdots,\bar{d}_{n-1}]\boldsymbol{p}\bar{\boldsymbol{d}}\\
&= \sigma_\omega^2\sum_{i=1}^{n-1} \bar{d}_i\bar{d}_{i+1} + \sigma_\omega^2\Big[-\sum_{i=1}^{n-1}\bar{d}_i^2, -\sum_{i=1}^{n-2}\bar{d}_i\bar{d}_{i+1},\cdots, -\sum_{i=1}^{1}\bar{d}_i\bar{d}_{i+n-2},0,\cdots,0\Big]\boldsymbol{\theta}
\end{aligned} \tag{4.114a}$$

$$\begin{aligned}
&E\{y(k-2)e(k)\}\\
&= \sigma_\omega^2[0,0,\bar{d}_1,\bar{d}_2,\cdots,\bar{d}_{n-2}]\boldsymbol{p}\bar{\boldsymbol{d}}\\
&= \sigma_\omega^2\sum_{i=1}^{n-2} \bar{d}_i\ \bar{d}_{i+2} + \sigma_\omega^2\Big[-\sum_{i=1}^{n-1}\bar{d}_i\ \bar{d}_{i+1}, -\sum_{i=1}^{n-2}\bar{d}_i^2,\\
&\quad -\sum_{i=2}^{n-3}\bar{d}_i\ \bar{d}_{i+2},\cdots, -\sum_{i=1}^{1}\bar{d}_i\ \bar{d}_{i+n-3},0,\cdots,0\Big]\boldsymbol{\theta}
\end{aligned} \tag{4.114b}$$

$$\begin{aligned}
&E\{y(k-n+1)e(k)\}\\
&= \sigma_\omega^2[0,\cdots,0,\bar{d}_1]\boldsymbol{p}\bar{\boldsymbol{d}}\\
&= \sigma_\omega^2 d_1 d_n + \sigma_\omega^2\Big[-\sum_{i=1}^{1}\bar{d}_i\bar{d}_{i+n-2},\cdots, -\sum_{i=1}^{1}\bar{d}_i\bar{d}_{i-1}, -\sum_{i=1}^{1}\bar{d}_i^2,0,\cdots,0\Big]\boldsymbol{\theta}
\end{aligned} \tag{4.114c}$$

$$E\{y(k-n)e(k)\} = 0 \tag{4.114d}$$

由此可得

$$E\{\boldsymbol{h}(k)e(k)\} = \boldsymbol{\lambda} + \boldsymbol{M}\boldsymbol{\theta}_0 \tag{4.115}$$

$$\boldsymbol{\lambda} = \sigma_\omega^2\Big[\sum_{i=1}^{n-1}\bar{d}_i\bar{d}_{i+1},\quad \sum_{i=1}^{n-2}\bar{d}_i\bar{d}_{i+2},\quad \cdots,\quad \sum_{i=1}^{1}\bar{d}_i\bar{d}_{i+n-1}\Big]^{\mathrm{T}} \in \mathbf{R}^{2n+1} \tag{4.116a}$$

$$
\boldsymbol{M}=\sigma_{\omega}^{2}\begin{bmatrix}
-\sum_{i=1}^{n-1}\bar{d}_i^2 & -\sum_{i=1}^{n-2}\bar{d}_i\bar{d}_{i+1} & -\sum_{i=1}^{n-3}\bar{d}_i\bar{d}_{i+2} & \cdots & -\sum_{i=1}^{1}\bar{d}_i\bar{d}_{i+n-2} & 0 & \cdots & 0\\
 & -\sum_{i=1}^{n-2}\bar{d}_i^2 & -\sum_{i=1}^{n-3}\bar{d}_i\bar{d}_{i+1} & \cdots & -\sum_{i=1}^{1}\bar{d}_i\bar{d}_{i+n-3} & 0 & \cdots & 0\\
 & & -\sum_{i=1}^{n-3}\bar{d}_i^2 & & \vdots & 0 & \cdots & 0\\
 & & & \ddots & & & & \\
 & & & & -\sum_{i=1}^{1}\bar{d}_i^2 & 0 & \cdots & 0\\
 & & \text{(对称于上三角)} & & & 0 & \cdots & 0\\
 & & & & & & \ddots & \vdots\\
 & & & & & & & 0
\end{bmatrix}
$$

(4.116b)

式(4.116b)表明，参变量 $\boldsymbol{\lambda}$ 和 $\boldsymbol{M}$ 由噪声模型的参数向量 $\bar{\boldsymbol{d}}$ 组成。当 $\bar{\boldsymbol{d}}$ 已知时，参变量 $\boldsymbol{\lambda}$ 和 $\boldsymbol{M}$ 可惟一确定。

上述分析表明，当状态方程(4.101)的噪声模型参数向量 $\bar{d}$ 及噪声 $\omega(k)$、$s(k)$ 和 $w(k)$ 的方差 σ_{ω}^2、σ_s^2 和 σ_w^2 已知时，式(4.101)的待辨识参数 $\boldsymbol{\theta}$ 可利用梯度校正法来确定。然而，由于梯度校正法用于状态方程参数辨识时，对噪声特性的先验知识要求过高，这将限制它的使用范围。

4.4.2 差分方程的参数辨识

设一个 SISO 系统用差分方程描述如下

$$
\begin{aligned}
&y(k)+a_1y(k-1)+\cdots+a_ny(k-n)\\
&=b_1u(k-1)+\cdots+b_nu(k-n)+d_1\omega(k-1)+\cdots+d_n\omega(k-n)
\end{aligned}
\tag{4.117}
$$

式中，$d_i(i=1,2,\cdots,n)$ 是假定已知的噪声模型参数；a_i 和 $b_i(i=1,2,\cdots,n)$ 是未知待辨识的系统模型参数；输入输出变量 $u(k)$ 和 $y(k)$ 含有测量噪声，如式(4.103)所示。噪声 $\omega(k)$、$s(k)$ 和 $w(k)$ 的统计特性与 4.4.1 小节所给的条件相同。与方程(4.104)一样，式(4.117)也可化成第二类随机性梯度校正参数辨识问题，如式(4.106)和(4.107)所示。同理，利用梯度校正法来解决式(4.117)的参数辨识问题，需要确定 $\boldsymbol{T}_2=E\{\boldsymbol{h}(k)e(k)\}$ 的关系。类似于 4.4.1 小节中 $E\{\boldsymbol{h}(k)e(k)\}$ 的推导，可得 $\boldsymbol{T}_2$ 为

$$
\boldsymbol{T}_2 = \sigma_\omega^2 \begin{bmatrix} [d_2, d_3, \cdots, d_{n-1}, d_n, 0]\boldsymbol{p}^{-1}\boldsymbol{d} \\ [d_3, d_4, \cdots, d_n, 0, 0]\boldsymbol{p}^{-1}\boldsymbol{d} \\ \vdots \\ [d_n, 0, \cdots, 0]\boldsymbol{p}^{-1}\boldsymbol{d} \\ \boldsymbol{0} \end{bmatrix} \in \mathbf{R}^{2n\times 1} \tag{4.118}
$$

式中，矩阵 $\boldsymbol{P}$ 如式(4.105)所示；$\boldsymbol{d}$ 为已知的噪声模型参数。由于 $\boldsymbol{T}_2$ 不能化成 $\boldsymbol{\theta}_0$ 的线性函数，参变量 $\boldsymbol{\lambda}$ 和 $\boldsymbol{M}$ 无法确定。但是，如果式(4.118)中的 $\boldsymbol{P}$ 用 $\hat{\boldsymbol{p}} = \boldsymbol{p}|_{\hat{\boldsymbol{\theta}}(k)}$代替，则可直接利用式(4.74)算法来估计式(4.117)的模型参数 $\boldsymbol{\theta}$。若方程(4.117)的阶次 $n \leqslant 3$，则 $\boldsymbol{T}_2$ 可化成 $\boldsymbol{\theta}_0$ 的线性函数。这时又可利用式(4.108)估计模型参数 $\boldsymbol{\theta}$。比如，当 $n=3$ 时，有

$$
\boldsymbol{\lambda} = \sigma_\omega^2[d_1d_2 + d_2d_3, d_1d_3, 0, 0, 0, 0]^{\mathrm{T}}, \quad \boldsymbol{M} = \sigma_\omega^2 \begin{bmatrix} d_1d_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & & & & & \\ 0 & & & & & \\ 0 & & & \boldsymbol{0} & & \\ 0 & & & & & \\ 0 & & & & & \end{bmatrix} \tag{4.119}
$$

显然，梯度校正法用于差分方程的参数辨识时，要求噪声模型参数 $\boldsymbol{d}$ 及噪声$\omega(k)$、$s(k)$和 $w(k)$的方差 σ_ω^2、σ_s^2 和σ_w^2 必须是已知的。这将大大限制梯度校正法的使用范围。

4.5 随机逼近法

随机逼近法是一种颇受重视的参数估计方法，它属于梯度校正法。

4.5.1 随机逼近原理

考虑模型辨识问题

$$
z(k) = \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta} + e(k) \tag{4.120}
$$

式中，$e(k)$是均值为零的噪声。显然，这种模型的参数辨识问题可以通过极小化$e(k)$的方差来实现，即求参数 $\boldsymbol{\theta}$ 的估计值，使下列准则函数达到极小值

$$
J(\boldsymbol{\theta}) = \frac{1}{2}E\{e^2(k)\} = \frac{1}{2}E\{[z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}]^2\} \tag{4.121}
$$

该准则函数的一阶负梯度为

$$\left[-\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]^{\mathrm{T}}=E\{\boldsymbol{h}(k)[z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}]\} \tag{4.122}$$

令其梯度为零

$$E\{\boldsymbol{h}(k)[z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}]\}\mid_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}=\mathbf{0} \tag{4.123}$$

原则上说，由式(4.123)可以求得使 $J(\boldsymbol{\theta})=\min$ 的参数估计值 $\hat{\boldsymbol{\theta}}$。但是，因为 $e(k)$ 的统计特性不知道，因此实际上还是无法求出式(4.123)的解。如果式(4.123)左边的数学期望用平均值来近似，即将式(4.123)近似写成

$$\frac{1}{L}\sum_{k=1}^{L}\boldsymbol{h}(k)[z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}]=\mathbf{0} \tag{4.124}$$

则有

$$\hat{\boldsymbol{\theta}}=\left[\sum_{k=1}^{L}\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)\right]^{-1}\left[\sum_{k=1}^{L}\boldsymbol{h}(k)z(k)\right] \tag{4.125}$$

显然，这种近似使问题退化成最小二乘问题，式(4.125)就是第 3 章讨论的最小二乘解。下面研究式(4.123)的随机逼近解。

设 x 是标量，$y(k)$是对应的随机变量，$p(y\mid x)$是 x 条件下 y 的概率密度函数，则随机变量 y 关于 x 的条件数学期望为

$$E\{y\mid x\}=\int y\mathrm{d}p(y\mid x) \tag{4.126}$$

记作

$$h(x)=E\{y\mid x\} \tag{4.127}$$

它是 x 的函数，称作回归函数。

对于给定的 α，设方程

$$h(x)=E\{y\mid x\}=\alpha \tag{4.128}$$

具有惟一解。当 $h(x)$函数的形式及条件概率密度函数 $p(y\mid x)$都不知道时，求方程(4.128)的解析解是困难的，这时可以利用随机逼近法来求解。所谓的随机逼近法就是利用变量 $x_1,x_2,\cdots$及其对应的随机变量 $y(x_1),y(x_2),\cdots$，通过迭代计算，逐步逼近方程(4.128)的解。常用的迭代法有 Robbins-Monro 算法和 Kiefer-Wolfowitz 算法。

1. Robbins-Monro 算法

求解式(6.128)的 Robbins-Monro 算法为

$$x(k+1)=x(k)+\rho(k)[\alpha-y(x(k))] \tag{4.129}$$

式中，$y(x(k))$是对应于 $x(k)$的 y 值；$\rho(k)$称作收敛因子。如果收敛因子满足下列条件

$$\begin{cases}\rho(k)>0,\forall k;\lim\limits_{k\to\infty}\rho(k)=0\\ \sum\limits_{k=1}^{\infty}\rho(k)=\infty;\sum\limits_{k=1}^{\infty}\rho^2(k)<\infty\end{cases} \tag{4.130}$$

则 $x(k)$在均方意义下收敛于方程(7.128)的解。满足式(4.130)条件的收敛因子,最简单的有 $\rho(k)=\dfrac{1}{k}$或$\rho(k)=\dfrac{b}{k+a}$。

Wolfowitz 进一步证实,当

(1) $\int_{-\infty}^{\infty}[y-h(x)]^2\mathrm{d}p(y\mid x)<\infty$;

(2) $|h(x)|\leqslant c+d|x|(-\infty<x<\infty)$;

(3) $h(x)<a\ (x<x_0);h(x)>a\ (x>x_0)$;

(4) 对$0<\delta_1<\delta_2<\infty$的任意 δ_1 和 δ_2,存在

$$\inf_{\delta_1\leqslant|x-x_0|\leqslant\delta_2}|h(k)-a|>0 \tag{4.131}$$

时,Robbins-Monro 算法依概率 1 收敛于真值解 x_0,即

$$\mathrm{Prob}\{\lim_{k\to\infty}x(k)=x_0\}=1 \tag{4.132}$$

2. Kiefer-Wolfowitz 算法

Robbins-Monro 算法一般是用来求方程(4.128)的根。后来,Kiefer 和 Wolfowitz 引用它来确定回归函数 $h(x)$的极值。如果回归函数 $h(x)$存在极值,那么 $h(x)$取极值处的 x 使得$\dfrac{\mathrm{d}h(x)}{\mathrm{d}x}=0$。根据 Robbins-Monro 算法,Kiefer 和 Wolfowitz 给出了求回归函数 $h(x)$极值的迭代算法

$$x(k+1)=x(k)-\rho(k)\frac{\mathrm{d}y}{\mathrm{d}x}\Big|_{x(k)} \tag{4.133}$$

式中,如果收敛因子 $\rho(k)$满足 Robbins-Monro 算法条件,则 Kiefer-Wolfowitz 算法是收敛的,即 $x(k)$的收敛值将使 $h(x(k))$达到极值。

Kiefer-Wolfowitz 算法可以直接推广到多维的情况。考虑标量函数 $J(\boldsymbol{\theta})$的极值问题,如果$J(\boldsymbol{\theta})$在 $\hat{\boldsymbol{\theta}}$点上取得极值,那么求$\hat{\boldsymbol{\theta}}$的迭代算法为

$$\hat{\boldsymbol{\theta}}(k+1)=\hat{\boldsymbol{\theta}}(k)-\rho(k)\frac{\partial J(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big|_{\hat{\boldsymbol{\theta}}(k)} \tag{4.134}$$

如果收敛因子 $\rho(k)$满足式(4.130)的条件,那么$\hat{\boldsymbol{\theta}}(k)$在均方意义下收敛于真值 $\boldsymbol{\theta}_0$

$$\lim_{k\to\infty}E\{[\hat{\boldsymbol{\theta}}(k)-\boldsymbol{\theta}_0]^{\mathrm{T}}[\hat{\boldsymbol{\theta}}(k)-\boldsymbol{\theta}_0]\}=0 \tag{4.135}$$

Kiefer-Wolfowitz 算法是随机逼近法的基础[80]。

例 4.2 设回归函数

$$h(x)=\frac{1}{2}E\{[z(k)-x]^2\} \tag{4.136}$$

求 $h(x)$取极小值时的 x 值。

解 根据式(4.133),有

$$\hat{x}(k)=\hat{x}(k-1)+\rho(k)[z(k)-\hat{x}(k-1)] \tag{4.137}$$

取 $\rho(k)=\frac{1}{k}$,则

$$k\hat{x}(k)=(k-1)\hat{x}(k-1)+z(k) \tag{4.138}$$

设$\hat{x}(k)$和 $z(k)$的 z 变换分别为$\hat{X}(z)$和 $Z(z)$,并利用 z 变换公式

$$\begin{cases} kf(k)\leftrightarrow -z\dfrac{\mathrm{d}}{\mathrm{d}z}F(z) \\ \displaystyle\sum_{i=0}^{k} f(i)\leftrightarrow \frac{z}{z-1}F(z) \end{cases} \tag{4.139}$$

可得

$$\frac{\mathrm{d}}{\mathrm{d}z}\hat{X}(z)=\frac{1}{1-z}Z(z) \tag{4.140}$$

反变换后,有

$$\hat{x}(k)=\frac{1}{k}\sum_{i=0}^{k}z(i) \tag{4.141}$$

使得 $h(x)=\min$。式(4.141)表明,使 $h(x)$达到极小值的 x 为观测值的均值,这是符合物理意义的。

4.5.2 随机逼近参数估计方法

考虑模型(4.120)的参数辨识问题。设准则函数为

$$J(\boldsymbol{\theta})=E\{h(\boldsymbol{\theta},\boldsymbol{D}^k)\} \tag{4.142}$$

式中,$h(\cdot)$为某标量函数;D^k 表示 k 时刻以前的输入输出数据集合。显然,准则函数的一阶负梯度为

$$\left[-\frac{\partial J(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right]^{\mathrm{T}}=\left[E\left\{-\frac{\partial}{\partial\boldsymbol{\theta}}h(\boldsymbol{\theta},\boldsymbol{D}^k)\right\}\right]^{\mathrm{T}}=E\{\boldsymbol{q}(\boldsymbol{\theta},\boldsymbol{D}^k)\} \tag{4.143}$$

那么模型(4.120)的辨识问题归结成求解方程

$$E\{\boldsymbol{q}(\boldsymbol{\theta},\boldsymbol{D}^k)\}=\boldsymbol{0} \tag{4.144}$$

利用随机逼近原理,有

$$\hat{\boldsymbol{\theta}}(k)=\hat{\boldsymbol{\theta}}(k-1)+\rho(k)\boldsymbol{q}[\hat{\boldsymbol{\theta}}(k-1),\boldsymbol{D}^k] \tag{4.145}$$

式中，$\rho(k)$为收敛因子，必须满足式(4.130)的条件。如果 $J(\boldsymbol{\theta})$取式(4.121)作为准则函数，则式(4.145)写成

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \rho(k)\boldsymbol{h}(k)[z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1)] \tag{4.146}$$

该式是利用随机逼近法解决模型(4.120)辨识问题的基本公式。下面具体讨论差分方程的参数辨识问题。

考虑模型

$$A(z^{-1})y(k) = B(z^{-1})u(k) + v(k) \tag{4.147}$$

式中，$v(k)$是均值为零、方差为 σ_v^2 的不相关噪声；输入输出数据对应的测量值为

$$\begin{cases} x(k) = u(k) + s(k) \\ z(k) = y(k) + w(k) \end{cases} \tag{4.148}$$

式中，$s(k)$和 $w(k)$分别是均值为零、方差为 σ_s^2 和 σ_w^2 的不相关随机噪声，且 $v(k)$、$s(k)$、$w(k)$和 $u(k)$在统计上两两不相关；且

$$\begin{cases} A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a} \\ B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b} \end{cases} \tag{4.149}$$

模型(4.147)可化成最小二乘格式

$$z(k) = \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta} + e(k) \tag{4.150}$$

式中

$$\begin{cases} \boldsymbol{h}(k) = [-z(k-1), \cdots, -z(k-n_a), x(k-1), \cdots, x(k-n_b)]^{\mathrm{T}} \\ \boldsymbol{\theta} = [a_1, a_2, \cdots, a_{n_a}, b_1, b_2, \cdots, b_{n_b}]^{\mathrm{T}} \\ e(k) = A(z^{-1})w(k) - B(z^{-1})s(k) + v(k) \end{cases} \tag{4.151}$$

显然，噪声 $e(k)$具有如下特性

$$\begin{cases} E\{e(k)\} = 0 \\ E\{e(i)e(j)\} = \begin{cases} \text{有限值}, & |i-j| \leqslant n \\ 0, & |i-j| > n \end{cases}, \quad n = \max(n_a, n_b) \\ E\{\boldsymbol{h}(k)e(k)\} \neq \boldsymbol{0} \end{cases} \tag{4.152}$$

根据式(4.121)，取准则函数为

$$J(\boldsymbol{\theta}) = \frac{1}{2}E\{[z(k+n) - \boldsymbol{h}^{\mathrm{T}}(k+n)\boldsymbol{\theta}]^2\} \tag{4.153}$$

利用随机逼近原理，可得参数 $\boldsymbol{\theta}$ 估计值的 Isermann 随机逼近算法

$$\begin{cases} \hat{\boldsymbol{\theta}}(k+n) = \hat{\boldsymbol{\theta}}(k-1) + \rho(l)\boldsymbol{h}(k+n)[z(k+n) - \boldsymbol{h}^{\mathrm{T}}(k+n)\hat{\boldsymbol{\theta}}(k-1)] \\ k = 1, n+2, 2n+3, \cdots \end{cases} \tag{4.154}$$

式中，为了避免误差积累，所用的数据必须是互不相关的，或者说数据中所含的噪声 $e(k)$ 必须是统计独立的。根据式(4.152)，如果每隔 $n+1$ 时刻递推计算一次，则可满足这一要求。收敛因子 $\rho(l)$ 必须满足式(4.130)的条件，自变量 l 可取 $l=k-1$ 或 $l=\dfrac{k-1}{n+1}$。一般说来，$\rho(l)$ 随着 k 的增加要有足够的下降速度，但 $\rho(l)$ 不能下降得太快，否则被处理的数据总量太少。Isermann 曾推荐一种选择 $\rho(l)$ 的方法，如图 4.12 所示。

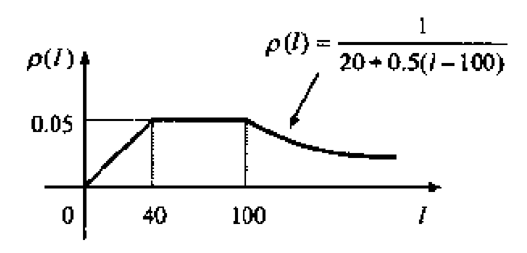


图 4.12　收敛因子 $\rho(l)$ 的选择

值得注意的是，式(4.154)算法所获得的参数估计值是有偏的。因为，根据式(4.123)，由准则函数(4.153)可得

$$
\begin{aligned}
\hat{\boldsymbol{\theta}} &= [E\{\boldsymbol{h}(k+n)\boldsymbol{h}^{\mathrm{T}}(k+n)\}]^{-1}E\{\boldsymbol{h}(k+n)z(k+n)\} \\
&= \boldsymbol{\theta}_0 + [E\{\boldsymbol{h}(k+n)\boldsymbol{h}^{\mathrm{T}}(k+n)\}]^{-1}E\{\boldsymbol{h}(k+n)e(k+n)\} \neq \boldsymbol{\theta}_0
\end{aligned}
\tag{4.155}
$$

式中

$$
E\{\boldsymbol{h}(k+n)e(k+n)\} = -\left[\begin{array}{c|c} \sigma_w^2 \boldsymbol{I}_{n_a} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \sigma_s^2 \boldsymbol{I}_{n_b} \end{array}\right]\boldsymbol{\theta}_0 \tag{4.156}
$$

可见，式(4.154)算法是有偏估计。如果把式(4.156)这项偏差引入算法，则可得到一种修正的随机逼近算法——相良节夫随机逼近法。

$$
\begin{cases}
\hat{\boldsymbol{\theta}}(k+n) = \hat{\boldsymbol{\theta}}(k-1) + \rho(l)\Bigg\{\boldsymbol{h}(k+n)[z(k+n) - \boldsymbol{h}^{\mathrm{T}}(k+n)\hat{\boldsymbol{\theta}}(k-1)] \\
\qquad + \left[\begin{array}{c|c} \sigma_w^2 \boldsymbol{I}_{n_a} & \\ \hline \boldsymbol{0} & \sigma_s^2 \boldsymbol{I}_{n_b} \end{array}\right]\hat{\boldsymbol{\theta}}(k-1)\Bigg\} \\
k = 1, n+2, 2n+3, \cdots
\end{cases}
\tag{4.157}
$$

式中，等式右边大括号中的第二项是为矫正有偏估计而引进的。式(4.157)在均方意义下是一致收敛的，即

$$\lim_{k\to\infty} E\{[\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}(k+n)]^{\mathrm{T}}[\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}(k+n)]\} = 0 \tag{4.158}$$

下面给出了一个仿真例子以比较算法(4.154)和(4.157)的辨识结果。

例 4.3　考虑如下仿真对象

$$\begin{cases} y(k) + 0.18y(k-1) - 0.784y(k-2) + 0.656y(k-3) = v(k) \\ z(k) = y(k) + w(k) \end{cases} \tag{4.159}$$

式中，$v(k)$和 $w(k)$分别是均值为零、方差为 1 和 0.25 的不相关随机噪声。采用算法(4.154)和(4.157)，得到的参数估计值误差如图 4.13 所示。算法(4.157)的辨识结果显然优于算法(4.154) 的辨识结果。

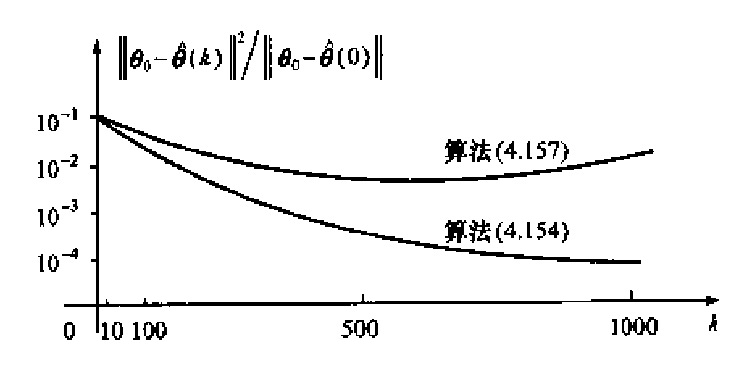


图 4.13　参数估计值误差的收敛情况

4.5.3　随机牛顿法

上面所讨论的随机逼近法实质上就是沿着准则函数的一阶负梯度方向去搜索极小点，其作用与最速下降法基本一致。式(4.145)就是这种思想的数学表现，它也可以写成

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) - \rho(k)\left[\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]^{\mathrm{T}}\Bigg|_{\hat{\boldsymbol{\theta}}(k-1)} \tag{4.160}$$

但是，当搜索点接近极小值点时，这种算法的收敛速度变得很慢。因此要想获得比较高的辨识精度，辨识时间将很长。为了加快收敛速度，可采用如下的牛顿法

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) - \left[\frac{\partial^2 J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right]^{-1}\left[\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]^{\mathrm{T}}\Bigg|_{\hat{\boldsymbol{\theta}}(k-1)} \tag{4.161}$$

式中，$\dfrac{\partial^2 J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}$表示准则函数 $J(\boldsymbol{\theta})$关于 $\boldsymbol{\theta}$ 的二阶导数，通常称之为 Hessian 矩阵。显然，Hessian 矩阵是对称阵，而且在递推计算过程中必须保证它的正定性，以使搜索方向始终指向“下山”方向。

一般说来，如果准则函数是确定性函数，那么牛顿算法(4.161)将有较快的收敛速度和较好的辨识精度。但是，这里所用的准则函数一般是回归函数，如式(4.142)所示，牛顿算法基本上不能适用，而且 Hessian 矩阵的表达式也难以求得。这就导致了随机牛顿算法

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \rho(k)\boldsymbol{R}^{-1}(k)\boldsymbol{q}(\boldsymbol{\theta},\boldsymbol{D}^k)\Big|_{\hat{\boldsymbol{\theta}}(k-1)} \quad (4.162)$$

式中，$\boldsymbol{q}(\boldsymbol{\theta},\boldsymbol{D}^k)$的定义见式(4.143)；$\boldsymbol{R}(k)$是 Hessian 矩阵在$\hat{\boldsymbol{\theta}}(k-1)$点上的近似形式，在特定的准则函数下，它可以再次用随机逼近法来确定。下面，利用随机牛顿法来讨论模型(4.120)的辨识问题，其准则函数取式(4.121)。根据式(4.143)，有

$$\boldsymbol{q}(\boldsymbol{\theta},\boldsymbol{D}^k) = \boldsymbol{h}(k)[z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}] \quad (4.163)$$

且 Hessian 矩阵为

$$\frac{\partial^2 J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} = E\{\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k)\} \quad (4.164)$$

显然，Hessian 矩阵是个回归函数，其准确表达式难以确定。设 $\boldsymbol{R}(k)$是 Hessian 矩阵在 k 时刻的估计值，则有

$$E\{\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k) - \boldsymbol{R}(k)\} = \boldsymbol{0} \quad (4.165)$$

根据罗 Robbins-Monro 算法(4.129)，可得 $\boldsymbol{R}(k)$的随机逼近算法为

$$\boldsymbol{R}(k) = \boldsymbol{R}(k-1) + \rho(k)[\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k) - \boldsymbol{R}(k-1)] \quad (4.166)$$

于是，模型(4.120)的随机牛顿算法 SNA(stochastic newton algorithm)归结为

$$\begin{cases}\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \rho(k)\boldsymbol{R}^{-1}(k)\boldsymbol{h}(k)[z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\,\hat{\boldsymbol{\theta}}(k-1)] \\ \boldsymbol{R}(k) = \boldsymbol{R}(k-1) + \rho(k)[\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k) - \boldsymbol{R}(k-1)]\end{cases} \quad (4.167)$$

式中，$\rho(k)$为收敛因子。

随机牛顿算法是一种很有用的算法。以它为基础，可以揭示许多辨识算法的内在联系，从而可以导出递推辨识算法的一般结构。比如，最小二乘递推算法(3.61)实际上已经包括在式(4.167)之中了。当取 $\rho(k)=\frac{1}{k}$，$\boldsymbol{R}(k)=\frac{1}{k}\boldsymbol{P}^{-1}(k)$时，式(3.61)可直接由式(4.167)导出。

梯度校正法，包括随机逼近法，主要特点是计算简单，可用于在线实时辨识。

4.6 小　结

梯度校正参数辨识的基本思想是，从给定的初始值$\hat{\boldsymbol{\theta}}(0)$开始，沿着准则函数

$J(\boldsymbol{\theta})$的负梯度方向修正模型参数的估计值$\hat{\boldsymbol{\theta}}(k)$，直至准则函数$J(\boldsymbol{\theta})$达到最小值。在确定性问题的梯度校正参数估计中，权矩阵的选择是至关重要的。若选取 Lyapunov 最佳权矩阵，则参数估计值将以最快的速度收敛于真值 $\boldsymbol{\theta}_0$。

根据输入向量 $\boldsymbol{h}(k)$、测量噪声 $\omega(k)$及被辨识参数$\hat{\boldsymbol{\theta}}(k)$之间是否相关，随机性问题的梯度校正参数辨识问题被分为三类。状态方程(4.101)的参数辨识可化为第二类随机性梯度校正参数辨识问题，应用定理 4.3 可得到参数的渐进无偏估计。应该注意的是梯度校正法用于状态方程和差分方程的参数辨识时，对噪声特性的先验知识要求很高。

随机逼近法是利用变量 $x_1,x_2,\cdots$及其对应的随机变量 $y(x_1),y(x_2),\cdots$，通过迭代计算，逐步逼近式(4.128)的解。常用的迭代算法有 Robbins-Monro 算法和 Kiefer-Wolfowitz 算法。随机逼近参数辨识是沿着准则函数的一阶负梯度方向去搜索极小点。牛顿逼近法是沿着准则函数的二阶负梯度方向去搜索极小点，因此其收敛速度较快。

习　题

1. 简述梯度校正参数辨识的基本原理。
2. 分析最小二乘类参数辨识算法与随机牛顿法之间的内在联系。
3. 某国家 1915～1955 年每隔 5 年的人口增长率如表 4.2。试用确定性梯度校正参数辨识算法求人口增长模型。设人口增长模型取 $y(t)=\theta_0+\theta_1 t+\theta_2 t^2$，权矩阵选用 $\boldsymbol{R}(k)=\boldsymbol{I}/\|h(t)\|^2$。

表 4.2

年代	1915	1920	1925	1930	1935	1940	1945	1950	1955
增长率	25.0%	23.7%	21.3%	18.9%	16.9%	17.9%	19.5%	23.6%	24.6%

4. 利用表 4.1 输入输出数据，根据确定性梯度校正参数辨识算法，用MATLAB软件求出该 SISO 系统的脉冲响应，要求被辨识参数的个数为 5 个。

第5章　极大似然法辨识方法

5.1　引　　言

极大似然法是现代辨识的参数估计方法之一。它是由 Fisher 发展起来的，其基本思想可追溯到高斯(1809年)。这种估计方法用于动态系统辨识，可以获得良好的估计效果。

除了相关分析法的古典辨识方法之外，前面已经讨论过两类现代辨识方法，一类是最小二乘方法，另一类是梯度校正法。它们不仅计算简单，而且参数估计量具有许多优良的统计性质，对噪声特性的先验知识要求也不高。本章主要讨论极大似然辨识方法，这类辨识方法的基本思想与前两类方法完全不同。对于极大似然法来说，需要构造一个以测量数据和未知参数有关的似然函数，并通过极大化这个函数获得模型的参数辨识。据此，极大似然法通常要求具有能够写出输出量的条件概率密度函数的先验知识，因而，计算工作量较大。但是，极大似然参数估计方法可以对具有有色噪声的系统模型进行辨识，在动态系统辨识中有着广泛的应用。它和最小二乘法以及预报误差方法存在着一定的联系。本章首先介绍极大似然参数辨识原理；其次讨论动态系统模型参数的极大似然估计，其中包括系统动态模型及噪声模型的分类与特点、极大似然估计与最小二乘估计的关系、协方差阵未知时的极大似然参数估计；最后，讨论递推的极大似然参数估计，其中包括极大似然递推算法的原理及方法、对开发的似然递推法辨识 MATLAB 仿真程序进行了剖析。

5.2　极大似然参数辨识原理

设 z 是一个随机变量，在参数 θ 条件下 z 的概率密度函数为 $p(z|\theta)$，z 的 L 个观测值构成一个随机序列 $\{z(k)\}$。如果把这 L 个观测值记作 $\mathbf{z}_L=[z(1),z(2),\cdots,z(L)]^{\mathrm{T}}$，则 $\mathbf{z}_L$ 的联合概率密度为 $p(\mathbf{z}_L|\theta)$，那么 θ 的极大似然估计就是使 $p(\mathbf{z}_L|\theta)|_{\hat{\theta}_{\mathrm{ML}}}=\max$ 的参数估计值，即有

$$\left[\frac{\partial p(\mathbf{z}_L\mid\theta)}{\partial\theta}\right]^{\mathrm{T}}_{\hat{\theta}_{\mathrm{ML}}}=\mathbf{0}\tag{5.1}$$

或

$$\left[\frac{\partial\ln p(\mathbf{z}_L\mid\theta)}{\partial\theta}\right]^{\mathrm{T}}_{\hat{\theta}_{\mathrm{ML}}}=\mathbf{0}\tag{5.2}$$

上式表示对联合概率密度 $p(z_L|\theta)$取自然对数,然后对参数 θ 求导。显然,对一组确定的数据 z_L,$p(z_L|\theta)$只是参数 θ 的函数,已不再是概率密度函数了。这时的 $p(z_L|\theta)$称作 θ 的函数,有时记作 $L(z_L|\theta)$。可见,概率密度函数和似然函数物理含义不同,但它们的数学表达式相同,即 $L(z_L/\theta)=p(z_L|\theta)$。因此极大似然原理又可表述成

$$\left[\frac{\partial L(z_L \mid \theta)}{\partial \theta}\right]^{\mathrm{T}}_{\hat{\theta}_{\mathrm{ML}}} = \mathbf{0} \tag{5.3}$$

或

$$\left[\frac{\partial \ln L(z_L \mid \theta)}{\partial \theta}\right]^{\mathrm{T}}_{\hat{\theta}_{\mathrm{ML}}} = \mathbf{0} \tag{5.4}$$

式中,$\ln L(z_L|\theta)$称作对数似然函数;$\hat{\theta}_{\mathrm{ML}}$称作极大似然参数估计值,它使得似然函数或对数似然函数达到最大值。式(5.3)或(5.4)就是极大似然函数原理的数学表示。它们的物理意义是:对一组确定的随机序列 z_L,设法找到参数估计值 $\hat{\theta}_{\mathrm{ML}}$,使得随机变量 z 在 $\hat{\theta}_{\mathrm{ML}}$条件下的概率密度函数最大可能地逼近随机变量 z 在 θ_0(真值)条件下的概率密度函数,即应有

$$p(z_L \mid \hat{\theta}_{\mathrm{ML}}) \xrightarrow{\max} p(z \mid \theta_0) \tag{5.5}$$

上式含义是指当 $p(z|\theta)$取极大值时,对应的估值$\hat{\theta}_{\mathrm{ML}}$才和真值 θ_0 误差最小,因此,它反映了极大似然原来的本质,但是数学上不好实现。可以证明式(5.3)或(5.4)是实现式(5.5)的数学形式,下面论述这一事实。

设 $z(1),z(2),\cdots,z(L)$是一组在独立观测条件下获得的随机序列,或者说是母集的一组互相独立的样本,那么随机变量 z 在参数θ 条件下的似然函数为

$$\begin{aligned} L(z_L \mid \theta) &= p(z(1) \mid \theta)\, p(z(2) \mid \theta) \cdots p(z(L) \mid \theta) \\ &= \prod_{k=1}^{L} p(z(k) \mid \theta) \end{aligned} \tag{5.6}$$

对应的对数似然函数为

$$l(z_L \mid \theta) = \ln L(z_L \mid \theta) = \sum_{k=1}^{L} \ln p(z(k) \mid \theta) \tag{5.7}$$

平均对数似然函数为

$$\bar{l}(z_L \mid \theta) = \frac{1}{L}\sum_{k=1}^{L} \ln p(z(k) \mid \theta) \xrightarrow{L \to \infty} E\{\ln p(z \mid \theta)\} \tag{5.8}$$

同理,随机变量 z 在参数θ_0 条件下的平均对数似然函数为

$$\bar{l}(z_L \mid \theta_0) \xrightarrow{L \to \infty} E\{\ln p(z \mid \theta_0)\} \tag{5.9}$$

定义

$$I(\theta_0,\theta)\triangleq E\{\ln p(z|\theta_0)\}-E\{\ln p(z|\theta)\}=E\left\{\ln\frac{p(z|\theta_0)}{p(z|\theta)}\right\} \quad (5.10)$$

$I(\theta_0,\theta)$称为 Kullback-Leibler 信息测度。若令

$$x=\frac{p(z\mid\theta)}{p(z\mid\theta_0)} \quad (5.11)$$

则 $x>0$,并利用不等式 $\ln x\leqslant x-1$,于是有

$$\ln\frac{p(z\mid\theta)}{p(z\mid\theta_0)}\leqslant\frac{p(z\mid\theta)}{p(z\mid\theta_0)}-1 \quad (5.12)$$

因 $p(z|\theta_0)>0$,上述不等式两边同乘以 $p(z|\theta_0)$,再对 z 积分,则有

$$\int_{-\infty}^{\infty}p(z|\theta_0)\ln\frac{p(z\mid\theta)}{p(z\mid\theta_0)}\mathrm{d}z$$

$$\leqslant\int_{-\infty}^{\infty}p(z|\theta)\mathrm{d}z-\int_{-\infty}^{\infty}p(z|\theta_0)\mathrm{d}z \quad (5.13)$$

考虑到全概率为 1,上式写成

$$E\left\{\ln\frac{p(z\mid\theta)}{p(z\mid\theta_0)}\right\}\leqslant 0 \quad (5.14)$$

综合考虑式(5.10)和(5.14),有

$$I(\theta_0,\theta)\geqslant 0 \quad (5.15)$$

式(5.5)要求 $p(z|\theta)$取极大值,这就意味着 $I(\theta_0,\theta)$必须取极小值。根据 $I(\theta_0,\theta)$的定义,并考虑到对一个给定的随机变量 z,其概率密度函数 $p(z_L|\theta_0)$是确定的,显然,从式(5.10)可知,极小化 $I(\theta_0,\theta)$等价于极大化 $E\{\ln p(z|\theta)\}$,由于 $E\{\ln p(z|\theta)\}$与 $L(z_L|\theta)$之间存在单调的函数关系,所以极大化 $L(z_L|\theta)$或 $\ln L(z_L|\theta)$与极大化 $E\{\ln p(z|\theta)\}$是等效的。因此式(5.3)或(5.4)体现了极大似然原理的内在实质,它们是极大似然辨识的重要依据。

例 5.1 考虑一个独立同分布的随机过程$\{x(t)\}$,在参数 θ 条件下随机变量 x 的概率密度为$p(x|\theta)=\theta^2 x\mathrm{e}^{-\theta x}$,$\theta>0$,试用解析法求参数 θ 的极大似然估计。

解 设 $\boldsymbol{x}_L=[x(1),x(2),\cdots,x(L)]^{\mathrm{T}}$ 表示随机变量 x 的 L 个观测值的向量,那么随机变量 x 在参数θ 条件下的似然函数为

$$L(\boldsymbol{x}_L\mid\theta)=\prod_{k=1}^{L}p(z(k)\mid\theta)$$

$$=\theta^{2L}\prod_{k=1}^{L}x(k)\exp\left[-\theta\sum_{k=1}^{L}x(k)\right] \quad (5.16)$$

对应的对数似然函数为

$$l(\boldsymbol{x}_L\mid\theta)=\ln L(\boldsymbol{x}_L|\theta)=2L\ln\theta+\sum_{k=1}^{L}\ln x(k)-\theta\sum_{k=1}^{L}x(k) \quad (5.17)$$

根据式(5.4),则有

$$\left[\frac{\partial l(\boldsymbol{x}_L \mid \theta)}{\partial \theta}\right]_{\hat{\theta}_{ML}} = 2L\frac{1}{\hat{\theta}_{ML}} - \sum_{k=1}^{L} x(k) = 0 \tag{5.18}$$

从而可得

$$\hat{\theta}_{ML} = 2L/\sum_{k=1}^{L} x(k) \tag{5.19}$$

又由于

$$\left.\frac{\partial^2 \ln L(\boldsymbol{x}_L \mid \theta)}{\partial \theta^2}\right|_{\hat{\theta}_{ML}} = -\frac{2L}{\hat{\theta}_{ML}^2} < 0 \tag{5.20}$$

所以$\hat{\theta}_{ML}$使似然函数达到了最大值。因此$\hat{\theta}_{ML}$是参数θ的极大似然估计值。

5.3 动态系统模型参数的极大似然估计

5.3.1 动态模型描述

1. 动态系统模型

动态模型示意图如图5.1所示。设:$A(z^{-1}) = C(z^{-1})$, $n(k) = e(k)/C(z^{-1})$,则有

$$\begin{cases} A(z^{-1})z(k) = B(z^{-1})u(k) + e(k) \\ e(k) = D(z^{-1})v(k) \end{cases} \tag{5.21}$$

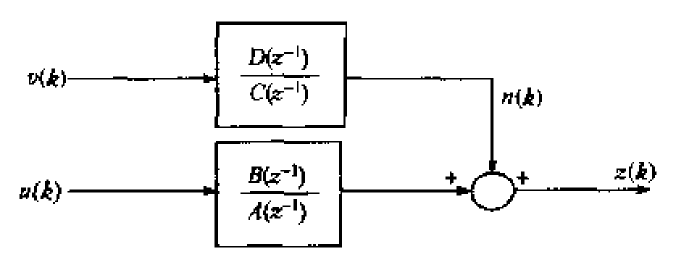


图5.1 动态模型示意图

式中,$v(k)$是均值为零、方差为σ_v^2、服从正态分布的不相关随机噪声;$u(k)$和$z(k)$表示系统的输入输出变量;且

$$\begin{cases} A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n} \\ B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n} \\ D(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_n z^{-n} \end{cases} \tag{5.22}$$

同时,设系统稳定,即$A(z^{-1})$和$D(z^{-1})$的所有零点都位于z平面的单位圆内,且$A(z^{-1})$、$B(z^{-1})$和$D(z^{-1})$没有公共因子,这意味过程是渐进稳定的。

2．噪声模型及其分类

根据第二章提到的表示定理，在一定条件下，图 5.1 中的有色噪声 $e(k)$ 可以由白噪声 $v(k)$ 驱动的线性环节的输出来表示。该线性环节叫做成形滤波器，如图 5.2 所示。

$v(k)$ → $D(z^{-1})/C(z^{-1})$ → $e(k)$

图 5.2　成形滤波器

从图 5.2 可直接写出噪声模型的脉冲传递函数，即

$$H(z^{-1}) = \frac{D(z^{-1})}{C(z^{-1})} \tag{5.23}$$

$v(k)$ 是白噪声，噪声模型式(5.23)直接决定 $e(k)$ 的噪声特点。在上式中，如果 $C(z^{-1})$ 或 $D(z^{-1})$ 简化为 1，则噪声模型的结构和特性也随之改变。根据其结构，噪声模型可分为以下三种类型：

(1) 自回归模型，简称 AR 模型，其模型结构为

$$C(z^{-1})e(k) = v(k) \tag{5.24}$$

(2) 平均滑动模型，简称 MA 模型，其模型结构为

$$e(k) = D(z^{-1})v(k) \tag{5.25}$$

(3) 自回归平均滑动模型，简称 ARMA 模型，其模型结构为

$$C(z^{-1})e(k) = D(z^{-1})v(k) \tag{5.26}$$

5.3.2　极大似然估计与最小二乘估计的关系

将模型式(5.21)写成

$$\boldsymbol{z}_L = \boldsymbol{H}_L\boldsymbol{\theta} + \boldsymbol{e}_L \tag{5.27}$$

式中

$$\begin{cases} \boldsymbol{z}_L = [z(1), z(2), \cdots, z(L)]^{\mathrm{T}} \\ \boldsymbol{e}_L = [e(1), e(2), \cdots, e(L)]^{\mathrm{T}} \\ \boldsymbol{\theta} = [a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n]^{\mathrm{T}} \\ \boldsymbol{H}_L = \begin{bmatrix} -z(0) & \cdots & -z(1-n) & u(0) & \cdots & u(1-n) \\ -z(1) & \cdots & -z(2-n) & u(1) & \cdots & u(2-n) \\ \vdots & & \vdots & \vdots & & \vdots \\ -z(L-1) & \cdots & -z(L-n) & u(L-1) & \cdots & u(L-n) \end{bmatrix} \end{cases} \tag{5.28}$$

因为

$$e(k) = v(k) + d_1 v(k-1) + \cdots + d_n v(k-n) \tag{5.29}$$

则有

$$\begin{cases} E\{e(k)e(k-j)\} = \sum_{l=0}^{n} d_l d_{l-j} \sigma_v^2 \\ d_0 \triangleq 1; \quad d_l = 0(l < 0 \text{ 或 } l > n) \end{cases} \tag{5.30}$$

噪声 $e(k)$ 的协方差阵记作

$$\boldsymbol{\Sigma}_e = E\{\boldsymbol{e}_L \boldsymbol{e}_L^{\mathrm{T}}\} = \begin{bmatrix} E\{e(1)e(1)\} & E\{e(1)e(2)\} & \cdots & E\{e(1)e(L)\} \\ E\{e(2)e(1)\} & E\{e(2)e(2)\} & \cdots & E\{e(2)e(L)\} \\ \vdots & \vdots & & \vdots \\ E\{e(L)e(1)\} & E\{e(L)e(2)\} & \cdots & E\{e(L)e(L)\} \end{bmatrix} \tag{5.31}$$

由于噪声 $v(k)$ 服从正态分布，系统的输出测量 $\boldsymbol{z}_L$ 也服从正态分布，即

$$\boldsymbol{z}_L \sim \mathcal{N}(\boldsymbol{H}_L\boldsymbol{\theta}, \boldsymbol{\Sigma}_e) \tag{5.32}$$

根据统计，正态分布的随机变量 x 的概率密度表达式为

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\bar{x})^2}{\sigma^2}\right] \tag{5.33}$$

式中，$\bar{x}$ 为正态分布函数中心值；σ^2 为尺度因子。则在参数 $\boldsymbol{\theta}$ 条件下，$\boldsymbol{z}_L$ 的概率密度函数可以写成

$$p(\boldsymbol{z}_L \mid \boldsymbol{\theta}) = (2\pi)^{-\frac{L}{2}} (\det \boldsymbol{\Sigma}_e)^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{z}_L - \boldsymbol{H}_L\boldsymbol{\theta})^{\mathrm{T}} \boldsymbol{\Sigma}_e^{-1} (\boldsymbol{z}_L - \boldsymbol{H}_L\boldsymbol{\theta})\right] \tag{5.34}$$

据此，对数似然函数可以写成

$$\begin{aligned} l(\boldsymbol{z}_L \mid \boldsymbol{\theta}) &= \ln L(\boldsymbol{z}_L \mid \boldsymbol{\theta}) = \ln p(\boldsymbol{z}_L \mid \boldsymbol{\theta}) \\ &= -\frac{L}{2}\ln 2\pi - \frac{1}{2}\ln\det\boldsymbol{\Sigma}_e - \frac{1}{2}(\boldsymbol{z}_L - \boldsymbol{H}_L\boldsymbol{\theta})^{\mathrm{T}} \boldsymbol{\Sigma}_e^{-1} (\boldsymbol{z}_L - \boldsymbol{H}_L\boldsymbol{\theta}) \end{aligned} \tag{5.35}$$

根据极大似然原理，将式(5.35)对 $\boldsymbol{\theta}$ 求导并令其等于零，可得

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = (\boldsymbol{H}_L^{\mathrm{T}} \boldsymbol{\Sigma}_e^{-1} \boldsymbol{H}_L)^{-1} \boldsymbol{H}_L^{\mathrm{T}} \boldsymbol{\Sigma}_e^{-1} \boldsymbol{z}_L \tag{5.36}$$

易验证对数似然函数的二阶导数小于零，即

$$\left.\frac{\partial^2 l(\boldsymbol{z}_L \mid \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}} < 0 \tag{5.37}$$

可见，式(5.36)的 $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$ 使对数似然函数 $l(\boldsymbol{z}_L \mid \boldsymbol{\theta})$ 取最大值。所以 $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$ 是 $\boldsymbol{\theta}$ 的极大似然估计。如果噪声 $e(t)$ 的协方差矩阵 $\boldsymbol{\Sigma}_e$ 已知，令 $\boldsymbol{\Sigma}_e = \sigma_e^2 \boldsymbol{I}$，$\boldsymbol{I}$ 为单位矩阵，$e(t)$ 是均值为零、方差为 σ_e^2 的不相关的随机噪声，那么由式(5.36)可直接写成

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = (\boldsymbol{H}_L^{\mathrm{T}} \boldsymbol{H}_L)^{-1} \boldsymbol{H}_L^{\mathrm{T}} \boldsymbol{z}_L \tag{5.38}$$

这时参数 $\boldsymbol{\theta}$ 的极大似然估计等价于最小二乘估计，但是前提是数据长度 L 应充分

大。否则,噪声方差会使辨识的精度受到影响。这一点可以由比较极大似然和最小二乘两种方法的噪声方差估计$\hat{\sigma}_e^2$加以证明。

5.3.3 协方差阵未知时的极大似然参数估计

在式(5.21)所表达的动态系统中,$v(k)$是服从正态分布的白噪声;$e(k)$是$v(k),v(k-1),\cdots,v(k-n)$的线性组合,是有色噪声,且$e(k)$的协方差阵$\boldsymbol{\Sigma}_e$未知。令

$$\boldsymbol{\theta}=[a_1,a_2,\cdots,a_n,b_1,b_2,\cdots,b_n,d_1,d_2,\cdots,d_n]^{\mathrm{T}} \tag{5.39}$$

在独立观测的前提下,当获得 L 组输入输出数据$\{u(k)\}$和$\{z(k)\}$后,在给定的参数$\boldsymbol{\theta}$和输入的$u(1),u(2),\cdots,u(L-1)$条件下,$z(1),z(2),\cdots,z(L)$的联合概率密度函数可写成

$$\begin{aligned}&p(z(1),z(2),\cdots,z(L)\mid u(1),u(2),\cdots,u(L-1),\boldsymbol{\theta})\\ =&p(z(L)\mid z(1),z(2),\cdots,z(L-1),u(1),u(2),\cdots,u(L-1),\boldsymbol{\theta})\\ &\times p(z(L-1)\mid z(1),z(2),\cdots,z(L-2),u(1),u(2),\cdots,u(L-1),\boldsymbol{\theta})\\ &\times\cdots\times p(z(1)\mid z(0),u(0),\boldsymbol{\theta})\\ =&\prod_{k=1}^{L}p(z(k)\mid z(1),z(2),\cdots,z(k-1),u(1),u(2),\cdots,u(k-1),\boldsymbol{\theta})\end{aligned} \tag{5.40}$$

根据模型(5.21),有

$$z(k)=-\sum_{i=1}^{n}a_iz(k-i)+\sum_{i=1}^{n}b_iu(k-i)+v(k)+\sum_{i=1}^{n}d_iv(k-i) \tag{5.41}$$

将式(5.41)代入到式(5.40),可得

$$\begin{aligned}&p(z(1),z(2),\cdots,z(L)\mid u(1),u(2),\cdots,u(L-1),\boldsymbol{\theta})\\ =&\prod_{k=1}^{L}p\Big(\Big[v(k)-\sum_{i=1}^{n}a_iz(k-i)+\sum_{i=1}^{n}b_iu(k-i)+\sum_{i=1}^{n}d_iv(k-i)\Big]\Big|\\ &z(1),z(2),\cdots,z(k-1),u(1),u(2),\cdots,u(k-1),\boldsymbol{\theta}\Big)\end{aligned} \tag{5.42}$$

当观测到 k 时刻,$(k-1)$时刻以前的$z(\cdot),u(\cdot),v(\cdot)$都已确定,且$v(k)$与$(k-1)$时刻以前的$z(\cdot),u(\cdot),\boldsymbol{\theta}$不相关,因此,上式可写成

$$\begin{aligned}&p(z(1),z(2),\cdots,z(L)\mid u(1),u(2),\cdots,u(L-1),\boldsymbol{\theta})\\ =&\prod_{k=1}^{L}p(v(k))+c=c+(2\pi)^{\frac{-L}{2}}(\sigma_v^2)^{\frac{-L}{2}}\exp\Big[-\frac{1}{2\sigma_v^2}\sum_{k=1}^{L}v^2(k)\Big]\end{aligned} \tag{5.43}$$

式中,c 为可由$(k-1)$时刻以前的确定量求出的常数。如果将 $\boldsymbol{z}_L$ 和$\boldsymbol{u}_{L-1}$写成

$$\begin{cases}\boldsymbol{z}_L=[z(1),z(2),\cdots,z(L)]^{\mathrm{T}}\\ \boldsymbol{u}_{L-1}=[u(1),u(2),\cdots,u(L-1)]^{\mathrm{T}}\end{cases} \tag{5.44}$$

那么观测值$\boldsymbol{z}_L$ 在$\boldsymbol{\theta}$ 和$\boldsymbol{u}_{L-1}$条件下的对数似然函数为

$$\begin{aligned} l(z_L \mid \boldsymbol{u}_{L-1}, \boldsymbol{\theta}) &= \ln L(z_L \mid \boldsymbol{u}_{L-1}, \boldsymbol{\theta}) = \ln p(z_L \mid \boldsymbol{u}_{L-1}, \boldsymbol{\theta}) \\ &= c - \frac{L}{2}\ln 2\pi - \frac{L}{2}\ln\sigma_v^2 - \frac{1}{2\sigma_v^2}\sum_{k=1}^{L} v^2(k) \end{aligned} \tag{5.45}$$

式中，$v(k)$满足下列关系

$$\begin{aligned} v(k) &= z(k) + \sum_{i=1}^{n} a_i z(k-i) \\ &\quad - \sum_{i=1}^{n} b_i u(k-i) - \sum_{i=1}^{n} d_i v(k-i) \end{aligned} \tag{5.46}$$

根据极大似然原理，噪声方差 σ_v^2 的极大似然估计$\hat{\sigma}_v^2$ 使得 $l(\boldsymbol{z}_L \mid \boldsymbol{u}_{L-1}, \boldsymbol{\theta})\Big|_{\hat{\sigma}_v^2} = \max$，即有

$$\left.\frac{\partial l(\boldsymbol{z}_L \mid \boldsymbol{u}_{L-1}, \boldsymbol{\theta})}{\partial \sigma_v^2}\right|_{\hat{\sigma}_v^2} = 0 \tag{5.47}$$

解得

$$\hat{\sigma}_v^2 = \frac{1}{L}\sum_{k=1}^{L} v^2(k) \tag{5.48}$$

把上式代入式(5.45)，则

$$\begin{aligned} l(\boldsymbol{z}_L \mid \boldsymbol{u}_{L-1}, \boldsymbol{\theta}) &= c - \frac{L}{2}\ln\frac{1}{L}\sum_{k=1}^{L} v^2(k) - \frac{L}{2} \\ &= c_1 - \frac{L}{2}\ln\frac{1}{L}\sum_{k=1}^{L} v^2(k) \end{aligned} \tag{5.49}$$

式中，$c_1 = c - L/2$。从极大似然原理知，参数 $\boldsymbol{\theta}$ 的极大似然估计$\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$必须使得 $l(\boldsymbol{z}_L \mid \boldsymbol{u}_{L-1}, \boldsymbol{\theta})\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}} = \max$，这等价于使得

$$V(\hat{\boldsymbol{\theta}}_{\mathrm{ML}}) = \frac{1}{L}\sum_{k=1}^{L} v^2(k)\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}} = \min \tag{5.50}$$

式中，$v(k)$满足式(5.46)的约束条件。

综上所述，当噪声 $e(k)$的协方差阵 $\boldsymbol{\Sigma}_e$ 未知时，模型式(5.21)的极大似然估计为：在式(5.46)的约束条件下，求参数 $\boldsymbol{\theta}$ 的极大似然估计$\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$必须使得 $V(\hat{\boldsymbol{\theta}}_{\mathrm{ML}}) = \min$。同时，噪声方差 σ_v^2 的估计值等于

$$\sigma_v^2 = \min V(\theta) = V(\hat{\boldsymbol{\theta}}_{\mathrm{ML}}) \tag{5.51}$$

显然，$V(\boldsymbol{\theta})$是参数 a_i，b_i 和 d_i 的函数，它关于 a_i，b_i 是线性的，而关于 d_i 却是非线性的。因此 $V(\boldsymbol{\theta})$的极小化问题不好求解，只能用迭代的方法求解。下面介绍

两种求 $V(\boldsymbol{\theta})$极小值的最优化迭代算法。

1．Lagrangian 乘子法

Lagrangian 乘子法是相良节夫等于 20 世纪 80 年代初提出来的一种基于极大似然思路的迭代算法。为了使对数似然函数式(5.45)达到最大，选目标函数为

$$V(\boldsymbol{\theta}) = \frac{1}{L}\sum_{k=1}^{L} v^2(k) \tag{5.52}$$

求目标函数的极小化，约束条件为

$$v(k) + \sum_{i=1}^{n} d_i v(k-i) - z(k) - \sum_{i=1}^{n} a_i z(k-i) + \sum_{i=1}^{n} b_i u(k-i) = 0 \tag{5.53}$$

引人 Lagrangian 乘子 $\lambda(k), k = n+1, n+2, \cdots, n+L$，构造如下的 Lagrangian 函数

$$\begin{aligned}\Im(\boldsymbol{\theta}) &= \frac{1}{L}\sum_{k=1}^{L} v^2(k) + \frac{1}{L}\sum_{k=1}^{L} \lambda(k)\Big[v(k) + \sum_{i=1}^{n} d_i v(k-i) \\ &\quad + \sum_{i=1}^{n} b_i u(k-i) - z(k) - \sum_{i=1}^{n} a_i z(k-i)\Big] \\ &= f(\boldsymbol{v}, \boldsymbol{\lambda}, a_i, b_i, d_i)\end{aligned} \tag{5.54}$$

把对目标函数 $V(\boldsymbol{\theta})$的极小化问题转化成对 Lagrangian 函数 $\Im(\boldsymbol{\theta})$的极小化问题。即变成了 Lagrangian 函数 $\Im(\boldsymbol{\theta})$分别对 $\boldsymbol{v}, \boldsymbol{\lambda}, \boldsymbol{\theta}(\boldsymbol{\theta} = [a_i, b_i, d_i]^{\mathrm{T}})$求极小值的问题。

(1) 首先取

$$\begin{cases}\left.\dfrac{\partial \Im(\boldsymbol{\theta})}{\partial v(j)}\right|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}} = \dfrac{2}{L}\hat{v}(j) + \dfrac{1}{L}\Big[\lambda(j) + \sum_{i=1}^{n} d_i \lambda(j+i)\Big] = 0 \\ j = n+1, n+2, \cdots, n+L\end{cases} \tag{5.55}$$

并令

$$\lambda(j) = 0, \quad j = L+1, L+2, \cdots, L+n \tag{5.56}$$

组成下列方程组

$$\begin{cases}\lambda(j) + \sum_{i=1}^{n} \hat{d}_i \lambda(j+i) + 2\hat{v}(j) = 0, \quad j = n+1, n+2, \cdots, L \\ \lambda(j) = 0, \quad j = L+1, L+2, \cdots, L+n\end{cases} \tag{5.57}$$

(2) 其次，Lagrangian 函数 $\Im(\boldsymbol{\theta})$对 $\lambda(k)$求导，并令之为零，可得

$$\hat{v}(k) = -\sum_{i=1}^{n} \hat{d}_i \hat{v}(k-i) + z(k) + \sum_{i=1}^{n} \hat{a}_i z(k-i) - \sum_{i=1}^{n} \hat{b}_i u(k-i) \tag{5.58}$$

(3) 由式(5.58)和(5.57)可求得 $\hat{v}(k)$ 和 $\lambda(k)$，求参数估计 $\hat{\boldsymbol{\theta}}_{ML}$ 值必须使 Lagrangian 函数 $\Im(\boldsymbol{\theta})$ 取极小值。由于 $\lambda(k)$ 和 $\hat{v}(k)$ 与 $\boldsymbol{\theta}$ 有关，对 $\boldsymbol{\theta}$ 不能以线性的形式进行估计，因此必须对

$$\begin{cases} \left.\dfrac{\partial \Im(\boldsymbol{\theta})}{\partial a_j}\right|_{\hat{\boldsymbol{\theta}}_{ML}} = -\dfrac{1}{L}\sum\limits_{k=n+1}^{L} \lambda(k) z(k-j) \\ \left.\dfrac{\partial \Im(\boldsymbol{\theta})}{\partial b_j}\right|_{\hat{\boldsymbol{\theta}}_{ML}} = \dfrac{1}{L}\sum\limits_{k=n+1}^{L} \lambda(k) u(k-j) \\ \left.\dfrac{\partial \Im(\boldsymbol{\theta})}{\partial d_j}\right|_{\hat{\boldsymbol{\theta}}_{ML}} = \dfrac{1}{L}\sum\limits_{k=n+1}^{L} \lambda(k)\, \hat{v}(k-j) \\ j = 1,2,\cdots,n \end{cases} \tag{5.59}$$

进行搜索，方可求得 $\hat{\boldsymbol{\theta}}_{ML}$，使 $\Im(\hat{\boldsymbol{\theta}}_{ML}) = \min$。搜索方法可采用 DFP(Davidon, Fletches, Powell)变尺度法，具体的搜索过程为

① 给定初始状态 $\hat{\boldsymbol{\theta}}_{ML}^{(i)}$ 和循环次数 M 等赋初值，对 n 阶系统，$\hat{\boldsymbol{\theta}}_{ML}^{(i)}$ 是 $3n\times 1$ 维矩阵，初始值可给随机小数阵；$\boldsymbol{H}^{(i)} = \boldsymbol{I}_{3n}$；赋 $i=0$。

② 根据 $\hat{\boldsymbol{\theta}}_{ML}^{(i)}$ 和 $\hat{v}(k)$ 的初值 $\hat{v}(1), \hat{v}(2), \cdots, \hat{v}(n)$，由式(5.58)计算 $\hat{v}(k)$，$k=n+1, n+2, \cdots, L$。

③ 根据所求的 $\hat{v}(k)$，由式(5.57)计算 $\lambda(k)$，$k=n+1, n+2, \cdots, L$。

④ 由式(5.59)Lagrangian 函数 $\Im(\boldsymbol{\theta})$ 关于 $\boldsymbol{\theta}$ 的梯度，记作

$$\boldsymbol{g}^{(i)} \triangleq \left[\frac{\partial \Im(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]^{\mathrm{T}}_{\hat{\boldsymbol{\theta}}_{ML}^{(i)}} \tag{5.60}$$

式中，$\boldsymbol{g}^{(i)}$ 为 $3n\times 1$ 维矩阵；$\boldsymbol{g}^{(i)}$ 中含有样本 $[-z(k-1), -z(k-2), u(k-1), u(k-2)]$ 和噪声估值 $\hat{v}(k)$ 的信息。

⑤ 令 $\boldsymbol{p}^{(i)} = -\boldsymbol{H}^{(i)}\boldsymbol{g}^{(i)}$。

⑥ 构造函数 $\phi^{(i)}(h^{(i)}) = \Im(\hat{\boldsymbol{\theta}}_{ML}^{(i)} + h^{(i)}\boldsymbol{p}^{(i)})$，用一维搜索法求 $h^{(i)}$ 使得

$$\phi^{(i)}(h^{(i)}) = \min$$

⑦ 令 $\hat{\boldsymbol{\theta}}_{ML}^{(i+1)} = \hat{\boldsymbol{\theta}}_{ML}^{(i)} + h^{(i)}\boldsymbol{p}^{(i)}$。

⑧ 重复第②、③、④步，根据 $\hat{\boldsymbol{\theta}}_{ML}^{(i+1)}$，计算 $\boldsymbol{g}^{(i+1)}$，并置 $\Delta\boldsymbol{g}^{(i)} = \boldsymbol{g}^{(i+1)} - \boldsymbol{g}^{(i)}$。

⑨ 利用 DFP 公式求 $\boldsymbol{H}^{(i+1)}$

$$\boldsymbol{H}^{(i+1)} = \boldsymbol{H}^{(i)} + h^{(i)}\frac{\boldsymbol{p}^{(i)}(\boldsymbol{p}^{(i)})^{\mathrm{T}}}{(\boldsymbol{p}^{(i)})^{\mathrm{T}}\Delta\boldsymbol{g}^{(i)}} - \frac{\boldsymbol{H}^{(i)}\Delta\boldsymbol{g}^{(i)}\Delta(\boldsymbol{g}^{(i)})^{\mathrm{T}}\boldsymbol{H}^{(i)}}{\Delta(\boldsymbol{g}^{(i)})^{\mathrm{T}}\boldsymbol{H}^{(i)}\Delta\boldsymbol{g}^{(i)}} \tag{5.61}$$

⑩ 如果 $\|\boldsymbol{g}^{(i)}\| \leqslant \varepsilon$ 或 $\|h^{(i)}\boldsymbol{p}^{(i)}\| \leqslant \varepsilon$，$\varepsilon$ 为指定的正小数，则意味着已搜索到参

数 $\boldsymbol{\theta}$ 的极大似然估计量 $\hat{\boldsymbol{\theta}}_{ML}$，停止计算；否则取上次计算中最后 n 个 $\hat{v}(k)$ 值作为下一次式(5.58)计算 $\hat{v}(k)$ 的初始值，并置 $i=i+1$，返回第②步，依此循环迭代，直至收敛或循环次数等于 M 中所赋的值为止。

DFP 变尺度法的程序流程图如图 5.3 所示。

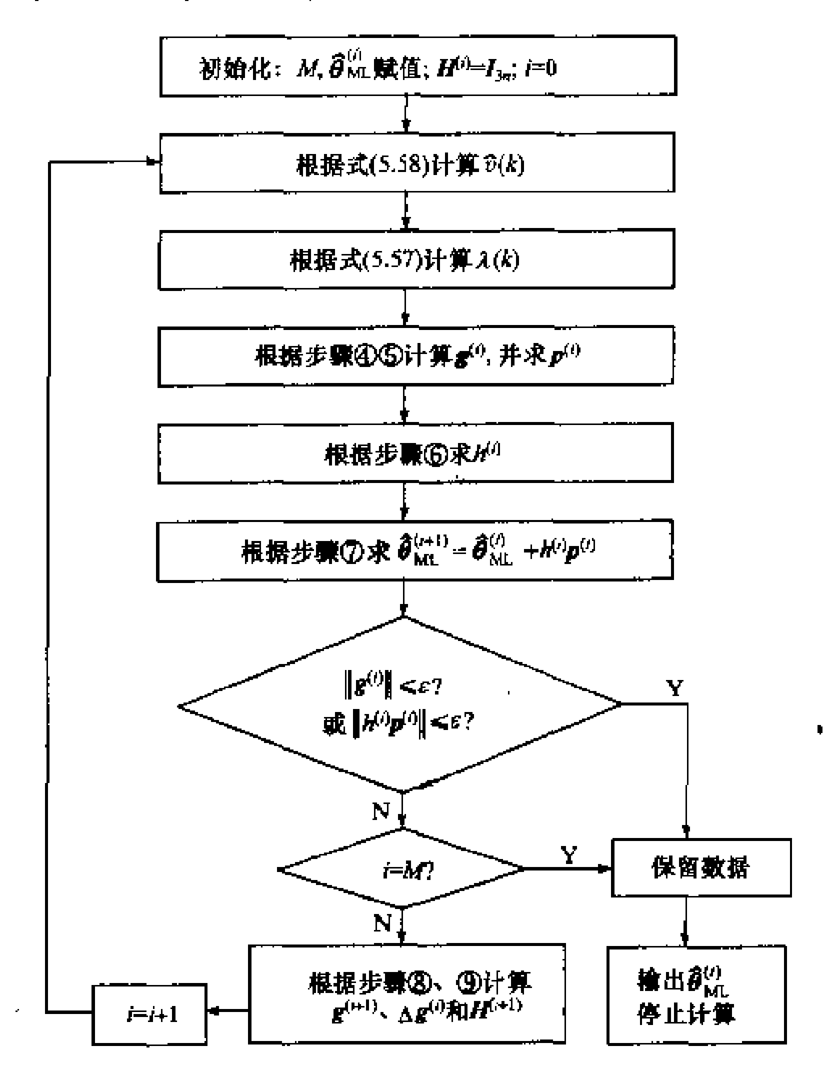


图 5.3　DFP 变尺度法的程序流程图

2. Newton-Raphson 法

现仍以式(5.52)和(5.53)分别为目标函数和约束条件，根据 Newton-Raphson 原理讨论对约束条件式(5.53)的极小化问题。设 $\hat{\boldsymbol{\theta}}_N$ 是利用第 N 批以前的输入输出数据 $\{u(k)\}$、$\{z(k)\}$ $(k=(N-1)L+1,(N-1)L+2,\cdots,NL)$ 求得的极大似然估计值，它使得

$$J_N(\boldsymbol{\theta}) = V_N(\boldsymbol{\theta}) = \frac{1}{L}\sum_{k=(N-1)L+1}^{NL} v^2(k) = \min \tag{5.62}$$

当又获得一批新的输入输出数据$\{u(k)\}$、$\{z(k)\}$($k=(NL+1,NL+2,\cdots,(N+1)L)$),使得

$$J_{N+1}(\boldsymbol{\theta})=\frac{1}{L}\sum_{k=NL+1}^{(N+1)L}v^2(k) \tag{5.63}$$

达到极小值。

根据 Newton-Raphson 原理,$\hat{\boldsymbol{\theta}}_{N+1}$与$\hat{\boldsymbol{\theta}}_N$ 满足下列递推关系

$$\hat{\boldsymbol{\theta}}_{N+1}=\hat{\boldsymbol{\theta}}_N-\boldsymbol{S}^{-1}\Big|_{\hat{\boldsymbol{\theta}}_N}\cdot\left[\frac{\partial J_{N+1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right]^{\mathrm{T}}_{\hat{\boldsymbol{\theta}}_N} \tag{5.64}$$

式中,$\boldsymbol{S}$ 为 Hessian 矩阵,定义为

$$\boldsymbol{S}\triangleq\frac{\partial^2 J_{N+1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2} \tag{5.65}$$

如果将式(5.63)写成递推形式

$$J_{N+1}(\boldsymbol{\theta},k)=J_{N+1}(\boldsymbol{\theta},k-1)+\frac{1}{2}v^2(k) \tag{5.66}$$

则可求得

$$\begin{aligned}\frac{\partial J_{N+1}(\boldsymbol{\theta},k)}{\partial\boldsymbol{\theta}}&=\frac{\partial J_{N+1}(\boldsymbol{\theta},k-1)}{\partial\boldsymbol{\theta}}+v(k)\frac{\partial v(k)}{\partial\boldsymbol{\theta}}\\&=\frac{\partial J_{N+1}(\boldsymbol{\theta},k-2)}{\partial\boldsymbol{\theta}}+\sum_{k=(N+1)L-1}^{(N+1)L}v(k)\frac{\partial v(k)}{\partial\boldsymbol{\theta}}=\cdots\\&=\sum_{k=NL+1}^{(N+1)L}v(k)\frac{\partial v(k)}{\partial\boldsymbol{\theta}}\end{aligned} \tag{5.67}$$

及

$$\begin{aligned}\frac{\partial^2 J_{N+1}(\boldsymbol{\theta},k)}{\partial\boldsymbol{\theta}^2}&=\frac{\partial^2 J_{N+1}(\boldsymbol{\theta},k-1)}{\partial\boldsymbol{\theta}^2}\\&\quad+\left[\frac{\partial v(k)}{\partial\boldsymbol{\theta}}\right]^{\mathrm{T}}\left[\frac{\partial v(k)}{\partial\boldsymbol{\theta}}\right]+v(k)\frac{\partial^2 v(k)}{\partial\boldsymbol{\theta}^2}\end{aligned} \tag{5.68}$$

忽略二阶导数项 $v(k)\dfrac{\partial^2 v(k)}{\partial\boldsymbol{\theta}^2}$,则有

$$\frac{\partial^2 J_{N+1}(\boldsymbol{\theta},k)}{\partial\boldsymbol{\theta}^2}\approx\frac{\partial^2 J_{N+1}(\boldsymbol{\theta},k-1)}{\partial\boldsymbol{\theta}^2}+\left[\frac{\partial v(k)}{\partial\boldsymbol{\theta}}\right]^{\mathrm{T}}\left[\frac{\partial v(k)}{\partial\boldsymbol{\theta}}\right] \tag{5.69}$$

因此

$$\begin{cases}S\Big|_{\hat{\boldsymbol{\theta}}_N}\approx\displaystyle\sum_{k=NL+1}^{(N+1)L}\left[\frac{\partial v(k)}{\partial\boldsymbol{\theta}}\right]^{\mathrm{T}}\left[\frac{\partial v(k)}{\partial\boldsymbol{\theta}}\right]\Big|_{\hat{\boldsymbol{\theta}}_N}\\ \left[\dfrac{\partial J_{N+1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right]^{\mathrm{T}}_{\hat{\boldsymbol{\theta}}_N}=\displaystyle\sum_{k=NL+1}^{(N+1)L}v(k)\left[\frac{\partial v(k)}{\partial\boldsymbol{\theta}}\right]^{\mathrm{T}}\Big|_{\hat{\boldsymbol{\theta}}_N}\end{cases} \tag{5.70}$$

将上式代入到式(5.64),可得到 Newton-Raphson 法获得$\hat{\boldsymbol{\theta}}_{N+1}$的递推法。上式中的$\left.\frac{\partial v(k)}{\partial \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_N}$展开后写成分别对参数$a_i, b_i, c_i$求一阶导数的形式,即

$$\begin{cases}\left.\dfrac{\partial v(k)}{\partial a_j}\right|_{\hat{\boldsymbol{\theta}}_N}=z(k-j)-\sum\limits_{i=1}^{n} d_i \left.\dfrac{\partial v(k-i)}{\partial a_j}\right|_{\hat{\boldsymbol{\theta}}_N} \\ \left.\dfrac{\partial v(k)}{\partial b_j}\right|_{\hat{\boldsymbol{\theta}}_N}=-u(k-j)-\sum\limits_{i=1}^{n} d_i \left.\dfrac{\partial v(k-i)}{\partial b_j}\right|_{\hat{\boldsymbol{\theta}}_N} \\ \left.\dfrac{\partial v(k)}{\partial d_j}\right|_{\hat{\boldsymbol{\theta}}_N}=-\hat{v}(k-j)-\sum\limits_{i=1}^{n} d_i \left.\dfrac{\partial v(k-i)}{\partial d_j}\right|_{\hat{\boldsymbol{\theta}}_N} \\ k=NL+n+1,\cdots,(N+1)L; \quad j=1,2,\cdots,n\end{cases} \tag{5.71}$$

下面给出 Newton-Raphson 法程序迭代具体步骤:

① 先采集一批输入输出数据$\{u(k)\}$、$\{z(k)\}$,利用最小二乘法获得参数的初步估值,作为极大似然参数估计的初始状态,记作$\hat{\boldsymbol{\theta}}_N$,对于 n 阶系统,$\hat{\boldsymbol{\theta}}_N$ 为 $3n\times1$ 维矩阵;给循环次数 M、允许迭代误差 ε 等赋值;$N=0$。

② 再采集一批长度为 L 的输入输出数据$\{u(k)\}$、$\{z(k)\}$$(k=1,2,\cdots,L)$,同时设定初始值$\hat{v}(1),\hat{v}(2),\cdots,\hat{v}(n)$和$\left.\frac{\partial v(k)}{\partial \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_N}(k=1,2,\cdots,n)$。

③ 根据式(5.58),利用$\hat{\boldsymbol{\theta}}_N$ 和$\hat{v}(k)$的初值,计算新的$\hat{v}(k)(k=NL+n+1, NL+n+2,\cdots,(N+1)L)$。

④ 根据式(5.71),利用$\hat{\boldsymbol{\theta}}_N$ 和$\left.\frac{\partial v(k)}{\partial \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_N}$初值,计算新的$\left.\frac{\partial v(k)}{\partial \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_N}$。

⑤ 利用所计算的新的$\hat{v}(k)$和$\left.\frac{\partial v(k)}{\partial \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_N}(k=NL+n+1,NL+n+2,\cdots,(N+1)L)$,根据式(5.70),计算 $J_{N+1}(\boldsymbol{\theta})$关于参数的梯度和 Hessian 矩阵 $\boldsymbol{S}$。

⑥ 根据式(5.64),递推计算$\hat{\boldsymbol{\theta}}_{N+1}$。

⑦ 取最后 n 个$\hat{v}(k)$和$\left.\frac{\partial v(k)}{\partial \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_N}$值,作为下一次迭代的初值;$N=N+1$。

⑧ 检测是否有$\left|\frac{\sigma_{N+1}^2-\sigma_N^2}{\sigma_N^2}\right|\leqslant\varepsilon$ 或$\|\hat{\boldsymbol{\theta}}_{N+1}-\hat{\boldsymbol{\theta}}_N\|\leqslant\varepsilon$ 成立,若是,保留$\hat{\boldsymbol{\theta}}_{N+1}$,停止迭代;若否,继续⑨。

⑨ 检测 $N=M$ 是否成立。若是,保留$\hat{\boldsymbol{\theta}}_{N+1}$,停止迭代;若否,转到③继续循环。

5.4　递推的极大似然参数估计

5.4.1　极大似然递推算法的原理及方法

实际上,Newton-Raphson 法是一种可以用于在线辨识递推算法。但这种方法每次需要采集一批长度为 L 的系统输入输出观测数据,即每批数据需要 L 次的观测,然后根据 L 次观测数据进行一次递推。本节将讨论一种每观测一次数据就递推计算一次参数估计值算法。本质上说,它只是一种近似的极大似然法。

考虑如下模型

$$A(z^{-1})z(k) = B(z^{-1})u(k) + D(z^{-1})v(k) \tag{5.72}$$

式中,$u(k)$和 $z(k)$是系统的输入、输出量;$\{v(k)\}$是均值为零、方差为 σ_v^2 的不相关随机噪声序列。且

$$\begin{cases} A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a} \\ B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b} \\ D(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_{n_d} z^{-n_d} \end{cases} \tag{5.73}$$

令

$$\boldsymbol{\theta} = [a_1, a_2, \cdots, a_{n_a}, b_1, b_2, \cdots, b_{n_b}, d_1, d_2, \cdots, d_{n_d}]^{\mathrm{T}} \tag{5.74}$$

则模型式(5.72)的参数极大似然问题就是求参数 $\boldsymbol{\theta}$,使得

$$J(\boldsymbol{\theta})\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}} = \frac{1}{2}\sum_{k=1}^{L} v^2(k)\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}} = \min \tag{5.75}$$

式中,$\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$为参数 $\boldsymbol{\theta}$ 的极大似然估计值;$v(k)$满足下列关系

$$v(k) = [D(z^{-1})]^{-1}[A(z^{-1})z(k) - B(z^{-1})u(k)] \tag{5.76}$$

如果 $v(k)$在$\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$点上进行泰勒展开,则可近似表示成

$$v(k) \approx v(k)\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}} + \frac{\partial v(k)}{\partial \boldsymbol{\theta}}\Big|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\mathrm{ML}}) \tag{5.77}$$

并设

$$\begin{aligned} \boldsymbol{h}_f(k) &\triangleq -\left[\frac{\partial v(k)}{\partial \boldsymbol{\theta}}\right]^{\mathrm{T}}\Bigg|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}} \\ &= -\left[\frac{\partial v(k)}{\partial a_1}, \cdots, \frac{\partial v(k)}{\partial a_{n_a}}, \frac{\partial v(k)}{\partial b_1}, \cdots, \frac{\partial v(k)}{\partial b_{n_b}}, \frac{\partial v(k)}{\partial d_1}, \cdots, \frac{\partial v(k)}{\partial d_{n_d}}\right]^{\mathrm{T}}\Bigg|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}} \end{aligned} \tag{5.78}$$

式中

$$\begin{cases}\left.\dfrac{\partial v(k)}{\partial a_j}\right|_{\hat{\theta}_{ML}} = [\hat{D}(z^{-1})]^{-1}z^{-j}z(k) \triangleq z^{-j}z_f(k) \\ \left.\dfrac{\partial v(k)}{\partial b_j}\right|_{\hat{\theta}_{ML}} = -[\hat{D}(z^{-1})]^{-1}z^{-j}u(k) \triangleq -z^{-j}u_f(k) \\ \left.\dfrac{\partial v(k)}{\partial d_j}\right|_{\hat{\theta}_{ML}} = -[\hat{D}(z^{-1})]^{-1}z^{-j}\hat{v}(k) \triangleq -z^{-j}\hat{v}_f(k)\end{cases} \tag{5.79}$$

式中，$z_f(k)$、$u_f(k)$和$\hat{v}_f(k)$分别表示 $z(k)$、$u(k)$和$\hat{v}(k)$的滤波值，满足下列关系

$$\begin{cases}z_f(k) = [\hat{D}(z^{-1})]^{-1}z(k) \\ u_f(k) = [\hat{D}(z^{-1})]^{-1}u(k) \\ \hat{v}_f(k) = [\hat{D}(z^{-1})]^{-1}\hat{v}(k)\end{cases} \tag{5.80}$$

或写成

$$\begin{cases}z_f(k) = z(k) - \hat{d}_1 z_f(k-1) - \cdots - \hat{d}_{n_d} z_f(k-n_d) \\ u_f(k) = u(k) - \hat{d}_1 u_f(k-1) - \cdots - \hat{d}_{n_d} u_f(k-n_d) \\ \hat{v}_f(k) = \hat{v}(k) - \hat{d}_1 \hat{v}_f(k-1) - \cdots - \hat{d}_{n_d} \hat{v}_f(k-n_d)\end{cases} \tag{5.81}$$

那么向量 $\boldsymbol{h}_f(k)$记作

$$\boldsymbol{h}_f(k) = [-z_f(k-1),\cdots,-z_f(k-n_a),u_f(k-1),\cdots,\\ u_f(k-n_b),\hat{v}_f(k-1),\cdots,\hat{v}_f(k-n_d)]^{\mathrm{T}} \tag{5.82}$$

为了得到极大似然估计的递推形式，先将 $J(\boldsymbol{\theta})$写成递推的形式

$$J(\boldsymbol{\theta},k) \approx J(\boldsymbol{\theta},k-1) + \frac{1}{2}v^2(k) \tag{5.83}$$

设$\hat{\boldsymbol{\theta}}(k-1)$是模型式(5.72)在 $k-1$ 时刻的极大似然估计值。若将 $J(\boldsymbol{\theta},k-1)$在$\hat{\boldsymbol{\theta}}(k-1)$点上进行泰勒级数展开，并将在该点上 $J(\boldsymbol{\theta},k-1)$对 $\boldsymbol{\theta}$ 的一阶导数忽略，则有

$$J(\boldsymbol{\theta},k) \approx \frac{1}{2}[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]^{\mathrm{T}}\boldsymbol{P}^{-1}(k-1)[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)] \\ + \frac{1}{2}\eta(k) + \frac{1}{2}v^2(k) \tag{5.84}$$

式中，$\eta(k)$是 $J(\boldsymbol{\theta},k-1)$进行泰勒级数展开时的残差项；同时

$$\boldsymbol{P}^{-1}(k-1)=\left.\frac{\partial^2 J(\boldsymbol{\theta},k-1)}{\partial \boldsymbol{\theta}^2}\right|_{\hat{\boldsymbol{\theta}}(k-1)} \tag{5.85}$$

是正定对称阵,令

$$J^*(\boldsymbol{\theta},k)=2J(\boldsymbol{\theta},k) \tag{5.86}$$

并注意到式(5.77)和(5.78),则

$$\begin{aligned}J^*(\boldsymbol{\theta},k)&\approx[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]^{\mathrm{T}}\boldsymbol{P}^{-1}(k-1)[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]+\eta(k-1)\\&\quad+\left\{v(k)\Big|_{\hat{\boldsymbol{\theta}}(k-1)}+\frac{\partial v(k)}{\partial\boldsymbol{\theta}}\Big|_{\hat{\boldsymbol{\theta}}(k-1)}[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]\right\}^2\\&=[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]^{\mathrm{T}}[\boldsymbol{P}^{-1}(k-1)+\boldsymbol{h}_f(k)\boldsymbol{h}_f^{\mathrm{T}}(k)][\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]\\&\quad-2v(k)\Big|_{\hat{\boldsymbol{\theta}}(k-1)}\boldsymbol{h}_f^{\mathrm{T}}(k)[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]+v^2(k)\Big|_{\hat{\boldsymbol{\theta}}(k-1)}+\eta(k)\end{aligned} \tag{5.87}$$

为了将式(5.87)配成二次型,设$\tilde{\boldsymbol{\theta}}(k-1)=\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)$,则有

$$J^*(\boldsymbol{\theta},k)\approx[\tilde{\boldsymbol{\theta}}(k-1)-\boldsymbol{r}(k)]^{\mathrm{T}}\boldsymbol{P}^{-1}(k)[\tilde{\boldsymbol{\theta}}(k-1)-\boldsymbol{r}(k)]+\eta^*(k) \tag{5.88}$$

式中

$$\begin{cases}\boldsymbol{P}^{-1}(k)=\boldsymbol{P}^{-1}(k-1)+\boldsymbol{h}_f(k)\boldsymbol{h}_f^{\mathrm{T}}(k)\\ \boldsymbol{r}(k)=\boldsymbol{P}(k)\boldsymbol{h}_f(k)v(k)\Big|_{\hat{\boldsymbol{\theta}}(k-1)}\triangleq\boldsymbol{K}(k)\hat{r}(k)\\ \eta^*(k)=\boldsymbol{r}^{\mathrm{T}}(k)\boldsymbol{P}^{-1}(k)\boldsymbol{r}(k)+v^2(k)\Big|_{\hat{\boldsymbol{\theta}}(k-1)}+\eta(k)\end{cases} \tag{5.89}$$

显然,$\eta^*(k)$大于零,故由式(5.88)可知,若 k 时刻的参数估计值$\hat{\boldsymbol{\theta}}(k)$使得

$$\tilde{\boldsymbol{\theta}}(k-1)\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}(k)}=\boldsymbol{r}(k)=\boldsymbol{K}(k)\hat{r}(k) \tag{5.90}$$

则$J^*(\boldsymbol{\theta},k)$可得最小值。利用矩阵反演公式

$$(\boldsymbol{A}+\boldsymbol{BC})^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{I}+\boldsymbol{CA}^{-1}\boldsymbol{B})^{-1}\boldsymbol{CA}^{-1} \tag{5.91}$$

则从式(5.89)第一式可推导出

$$\boldsymbol{P}(k)=\boldsymbol{P}(k-1)-\frac{\boldsymbol{P}(k-1)\boldsymbol{h}_f(k)\boldsymbol{h}_f^{\mathrm{T}}(k)\boldsymbol{P}(k-1)}{1+\boldsymbol{h}_f^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}_f(k)} \tag{5.92}$$

另外,类似于最小二乘法的推导,可获得增益矩阵的递推公式为

$$\boldsymbol{K}(k)=\boldsymbol{P}(k-1)\boldsymbol{h}_f(k)[1+\boldsymbol{h}_f^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}_f(k)]^{-1} \tag{5.93}$$

于是,递推的极大似然参数估计(RML)算法描述为式(5.94)。

式(5.94)表明,极大似然参数估计递推算法类似于增广最小二乘法,所不同的只是向量 $\boldsymbol{h}_f(k)$的构造不一样,极大似然参数估计递推算法的程序框图如图 5.4

所示。

$$
\begin{cases}
\hat{\boldsymbol{\theta}}(k)=\hat{\boldsymbol{\theta}}(k-1)+\boldsymbol{K}(k)\hat{v}(k)\\
\boldsymbol{K}(k)=\boldsymbol{P}(k-1)\boldsymbol{h}_f(k)\left[\boldsymbol{h}_f^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}_f(k)+\boldsymbol{I}\right]^{-1}\\
\boldsymbol{P}(k)=\left[\boldsymbol{I}-\boldsymbol{K}(k)\boldsymbol{h}_f^{\mathrm{T}}(k)\right]\boldsymbol{P}(k-1)\\
\hat{v}(k)=z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1)\\
\boldsymbol{h}(k)=[-z(k-1),\cdots,-z(k-n_a),u(k-1),\cdots,u(k-n_b),\\
\qquad\qquad \hat{v}(k-1),\cdots,\hat{v}(k-n_d)]^{\mathrm{T}}\\
\boldsymbol{h}_f(k)=[-z_f(k-1),\cdots,-z_f(k-n_a),u_f(k-1),\cdots,u_f(k-n_b),\\
\qquad\qquad \hat{v}_f(k-1),\cdots,\hat{v}_f(k-n_d)]^{\mathrm{T}}\\
z_f(k)=z(k)-\hat{d}_1(k)z_f(k-1)-\cdots-\hat{d}_{n_d}(k)z_f(k-n_d)\\
u_f(k)=u(k)-\hat{d}_1(k)u_f(k-1)-\cdots-\hat{d}_{n_d}(k)u_f(k-n_d)\\
\hat{v}_f(k)=\hat{v}(k)-\hat{d}_1(k)\hat{v}_f(k-1)-\cdots-\hat{d}_{n_d}(k)\hat{v}_f(k-n_d)
\end{cases}
\tag{5.94}
$$

式中,如果定义$\hat{d}_i(k)=0(i>n_d)$,则向量 $\boldsymbol{h}_f(k)$还可以写成如下的递推形式

$$
\begin{aligned}
&\boldsymbol{h}_f(k+1)\\
&=[z_f(k+1),u_f(k+1),\hat{v}_f(k+1)]^{\mathrm{T}}\\
&=\left[\begin{array}{cccc|cccc|cccc}
-d_1(k) & \cdots & & -d_{n_a}(k) & & & & & & & & \\
1 & & & \boldsymbol{0} & & \boldsymbol{0} & & & & \boldsymbol{0} & & \\
& \ddots & & & & & & & & & & \\
\boldsymbol{0} & & 1 & \boldsymbol{0} & & & & & & & & \\
\hline
& & & & -d_1(k) & \cdots & & -d_{n_d}(k) & & & & \\
& \boldsymbol{0} & & & 1 & & & \boldsymbol{0} & & \boldsymbol{0} & & \\
& & & & & \ddots & & & & & & \\
& & & & \boldsymbol{0} & & 1 & \boldsymbol{0} & & & & \\
\hline
& & & & & & & & -d_1(k) & \cdots & & -d_{n_d}(k) \\
& \boldsymbol{0} & & & & \boldsymbol{0} & & & 1 & & & \boldsymbol{0} \\
& & & & & & & & & \ddots & & \\
& & & & & & & & \boldsymbol{0} & & 1 & 0
\end{array}\right]
\left[\begin{array}{c}
-z_f(k-1)\\ -z_f(k-2)\\ \vdots\\ -z_f(k-n_a)\\ \hline
u_f(k-1)\\ u_f(k-2)\\ \vdots\\ u_f(k-n_b)\\ \hline
\hat{v}_f(k-1)\\ \hat{v}_f(k-2)\\ \vdots\\ \hat{v}_f(k-n_d)
\end{array}\right]\\
&\quad+\left[\begin{array}{c}
-z(k)\\ \boldsymbol{0}\\ \hline u(k)\\ \boldsymbol{0}\\ \hline \hat{v}(k)\\ \boldsymbol{0}
\end{array}\right]
\end{aligned}
\tag{5.95}
$$

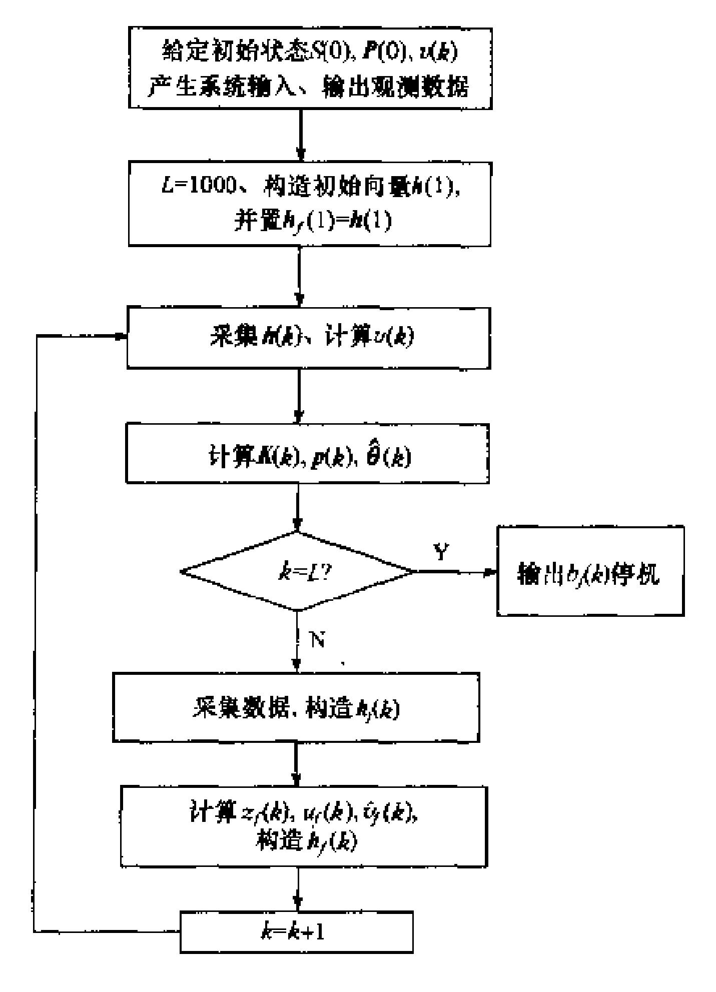


图 5.4　RML 辨识流程图

初始值一般取$\hat{\boldsymbol{\theta}}(0)=\boldsymbol{\varepsilon}$($\boldsymbol{\varepsilon}$ 为充分小的向量)，$\boldsymbol{P}(0)=\boldsymbol{I}$ 及$\hat{f}(k)=0, k=-1,-2,\cdots,-n_d$；并利用输入输出数据构造 $\boldsymbol{h}(1)$，使之不为全零向量，同时置 $\boldsymbol{h}_f(1)=\boldsymbol{h}(1)$。在这些初始状态下，利用式(5.94)，便可逐步递推计算 $\boldsymbol{K}(k)$、$\boldsymbol{P}(k)$和$\hat{\boldsymbol{\theta}}(k)$。

5.4.2　似然递推法辨识 MATLAB 仿真及程序剖析

例 5.2　系统模型如图 5.5 所示。试用递推的极大似然法求系统辨识的参数集 $\boldsymbol{\theta}$。图中 $v(k)$为随机信号，输入信号是幅值为 ±1 的 M 序列或随机信号，要求画出程序流程图，打印出程序(程序中带有注释)和辨识中的参数、误差曲线。

解　首先解释编程所用的部分字母：由于在 MATLAB 语言中无法用希腊字母描述，无法用上标及下标，故$\hat{\boldsymbol{\theta}}(k)$用“o”和“o1”表示；令 $\boldsymbol{P}(k)=\boldsymbol{P}_i(i=0,1)$；$\mathbf{K}(k)=\mathbf{K}$；产生 M 序列时，$a(i)$，$b(i)$，$c(i)$，$d(i)$表示四级移位寄存器的第 1，2，3，4 级寄存器的输出；

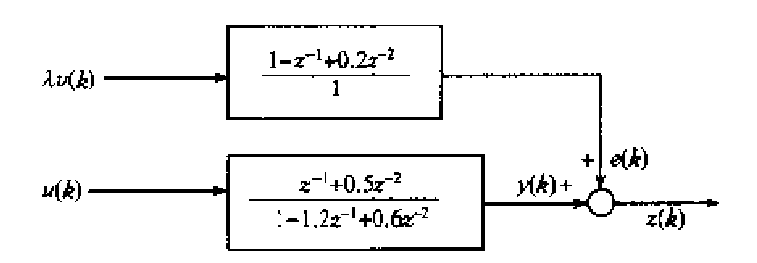


图 5.5　例 5.2 系统模型

1）编程如下

（光盘上该程序：FLch5RMLeg2.m，可在 MATLAB 6.I 下直接运行）：

```
clear       %清零
a(1)=1;b(1)=0;c(1)=1;d(1)=0;   %产生 m 序列的四级移位寄存器的初始值"1010"
u(1)=d(1);z(1)=0;z(2)=0;   %初始化
for    i=2:1200          %产生 m 序列 u(i)
  a(i)=xor(c(i-1),d(i-1));   %表示第 3 和第 4 寄存器的输出模 2 和作第 1 寄存器的输入
  b(i)=a(i-1);c(i)=b(i-1);d(i)=c(i-1);u(i)=d(i);
end    %产生 m 序列结束
u;   %若取去";"可以在程序运行中观测到 m 序列各个采样点上的值
v=randn(1200,1);%产生正态分布随机数 1200 个
V=0;      %计算噪声方差,V1 代表 σv²
for i=1:1200
    V=V+v(i)*v(i);
end
    V1=V/1200      %计算噪声方差结束
for k=3:1200      %根据 v 和 u 计算 z(k)
    z(k)=1.2*z(k-1)-0.6*z(k-2)+u(k-1)+0.5*u(k-2)+v(k)-v(k-1)+0.2*v(k-2);
end
    o1=0.001*ones(6,1);p0=eye(6,6);      %赋初值
    zf(1)=0.1;zf(2)=0.1;vf(2)=0.1;vf(1)=0.1;uf(2)=0.1;uf(1)=0.1;
for k=3:1200      %递推迭代计算数值和误差值开始
    h=[-z(k-1);-z(k-2);u(k-1);u(k-2);v(k-1);v(k-2)];
    hf=h;
    K=p0*hf*inv(hf'*p0*hf+1);
    p=[eye(6,6)-K*hf']*p0;      v(k)=z(k)-h'*o1;

    o=o1+K*v(k); p0=p; o1=o;  % o 代表 θ̂(k)
    a1(k)=o(1);a2(k)=o(2);b1(k)=o(3);b2(k)=o(4);%将 6 个参数分离

    d1(k)=o(5);d2(k)=o(6);% a1,a2,b1,b2,d1,d2 分别是 â1,â2,b̂1,b̂2,d̂1,d̂2 θ̂(k)
    e1(k)=abs(a1(k)+1.2);e2(k)=abs(a2(k)-0.6);e3(k)=abs(b1(k)-1.0);%计算误差
    e4(k)=abs(b2(k)-0.5);e5(k)=abs(d1(k)+1.0);e6(k)=abs(d2(k)-0.2);
    zf(k)=z(k)-d1(k)*zf(k-1)-d2(k)*zf(k-2);%构造 hf
```

```
    uf(k) = u(k) - d1(k) * uf(k - 1) - d2(k) * uf(k - 2);
    vf(k) = v(k) - d1(k) * vf(k - 1) - d2(k) * vf(k - 2);
    hf = [ - zf(k - 1); - zf(k - 2);
    uf(k - 1); uf(k - 2); vf(k - 1); vf(k - 2)];
end      %递推迭代计算结束

o1      %可以在程序运行中观察到参数 o1,o1 代表 θ̂
V1      %可以在程序运行中观察到噪声方差 V1,V1 代表 σv²
subplot(2,1,1)      %绘图
k = 1:1200;
plot(k,a1,'k:',k,a2,'b',k,b1,'r',k,b2,'m:',k,d1,'g',k,d2,'k');
xlabel('k'),  ylabel('parameter'),
title('The parameter idendification of the RML');
legend('a1 = - 1.2,','a2 = 0.6','b1 = 1.0','b2 = 0.5','d1 = - 1.0',' d2 = 0.2');    %图标注
end
subplot(4,1,2)
k = 1:1200;
plot(k,e1,'k',k,e2,'b',k,e3,'r',k,e4,'m',k,e5,'g',k,e6,'k');
xlabel('k'); ylabel('error'); % title(' 误差曲线 ')
end
subplot(4,1,3)
k = 1:1200;
plot(k,u); xlabel('k'); ylabel('input'); % title(' 系统激励信号:M序列 ')
end
subplot(4,1,4)
k = 1:1200;
plot(k,v); xlabel('k'); ylabel('random noise ); % title(' 系统所加的随机噪声 ')
end
```

2) 程序运行结果如图 5.6 所示

3) RML 辨识结果(估值)与真值的比较

RML 辨识结果(估值)与真值的比较如表 5.1 所示,显然在噪声方差 σ_v^2 接近 1 时,可以得到理想的辨识结果。

表 5.1 RML 法估值与真值的比较

参数估计值	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_1$	$\hat{b}_2$	$\hat{d}_1$	$\hat{d}_2$	σ_v^2
真值 / $k=2000$	−1.2	0.6	1.0	0.5	−1.0	0.2	1.0
估计值	−1.1892	0.5639	0.9548	0.4505	−0.9500	0.1873	0.9849

4) 程序调试注意事项

在编程序时,由于参数 $a_1=-1.2, d_1=-1.0$ 及 $a_2=0.6, b_2=0.5$ 分别相差

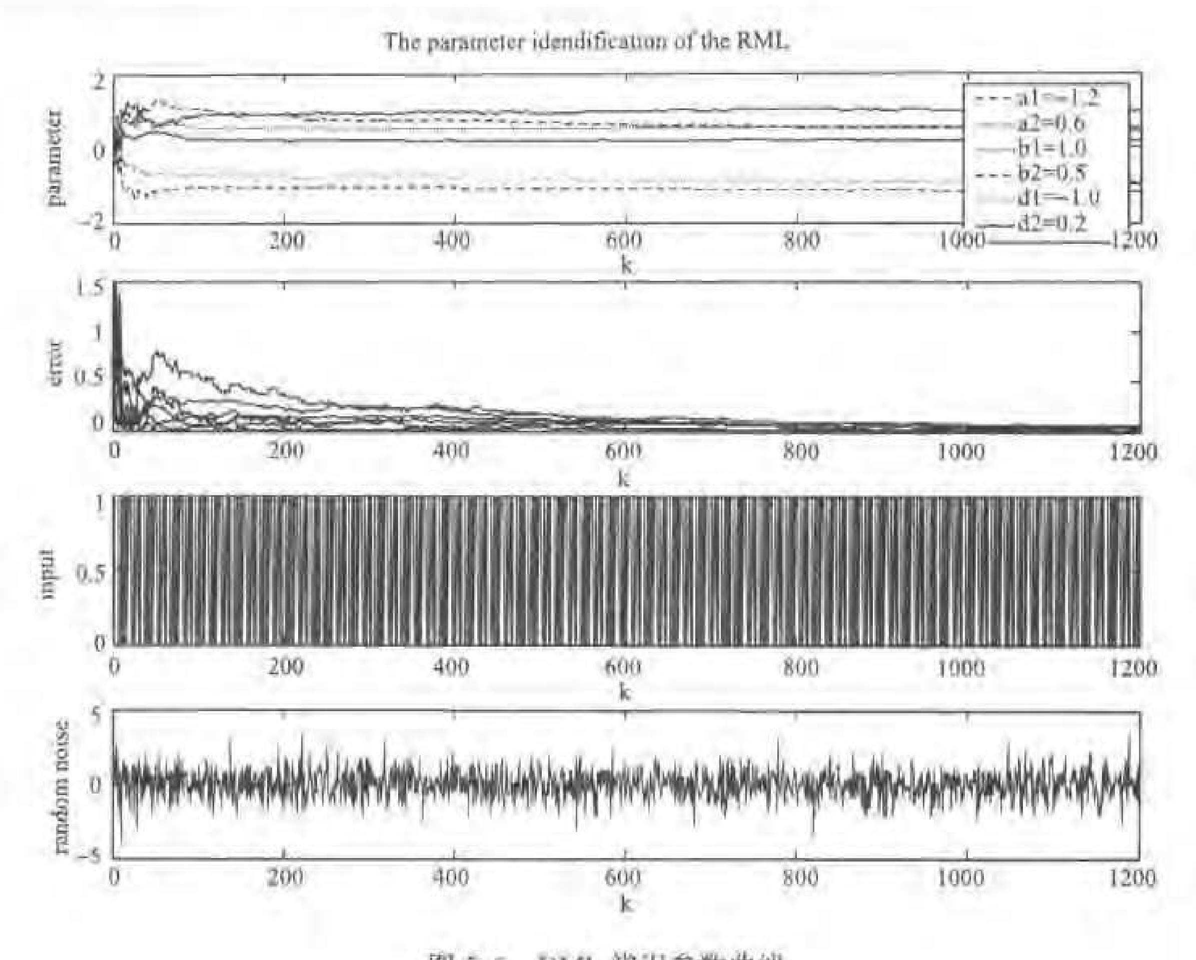


图 5.6　RML 辨识参数曲线

很小，故在绘图程序中用 plot(k,a1,'k:',k,a2,'b',k,b1,'r',k,b2,'m:',k,d1,'g',k,d2,'k')，即给 a_1 采用黑色的虚线 'k:' 以区别于 d_1，给 a_2 采用蓝色的实线 'b' 以区别于 b_2。由于程序产生的 1200 个正态分布随机数均值不是完全为零（或噪声方差等于 1），且每次运行都是随机的，因而，不是每次运行程序都能得到最理想的估值。在噪声方差最接近 1 时，可得到最接近真值的估值。所以应将程序多运行几次，在 MATLAB 的主界面上注意观察噪声方差 V_1（V_1 代表 σ_v^2）和参数 o_1（o_1 代表 $\hat{\theta}$），从而得到最接近真值的估值。另法：可在原程序的产生随机数（v = randn(1200,1);）和计算噪声方差结束（$V_1 = V/1200$）之间增加一段判断方差是否接近 1 的程序，若接近 1 程序继续运行，若否，另产生随机数，也可以帮助您得到理想的估值。

5.5　小　结

本章的主要内容是极大似然辨识方法，其中 5.1、5.2 节介绍了极大似然法的基本概念与极大似然参数辨识的原理，由条件概率密度函数 $p(z_L|\theta)$ 引出对数似

然函数 $\ln L(z_L|\theta)$，只要似然函数 $\ln L(z_L|\theta)$ 取极大值便可以使系统噪声方差最小。在 5.3 节动态系统模型的参数极大似然估计中，介绍了动态系统模型及噪声模型分类、极大似然和最小二乘法的关系、两种基于极大似然的迭代法，即 DFP 变尺度法和 Newton-Raphson 法。最后讨论了广泛应用的递推极大似然辨识法求系统模型的参数集 $\boldsymbol{\theta}$ 的方法和步骤，并针对具有噪声的二阶系统采用 MATLAB 进行了编程仿真，程序中加有详细的注释和调试程序的方法。

习　题

1. 简述噪声的模型及其分类。
2. 白噪声和有色噪声的区别是什么？
3. 极大似然参数估计与最小二乘估计主要区别是什么？
4. 系统模型与图 5.5 相类似，被辨识系统中参数的真值为：$a_1=-1.6, a_2=1.0, b_1=1.2, b_2=0.9, d_1=-0.6, d_2=0.2$，试用 DFP 变尺度法对系统的参数进行辨识，打印出辨识结果。提示：按照图 5.3 DFP 变尺度法的流程图，采用 MATLAB 语言编程。

第 6 章 离散随机系统的自适应滤波

在介绍了古典辨识、最小二乘辨识、梯度校正辨识、极大似然辨识之后，本章介绍离散随机系统的自适应滤波。本章前两节的内容为 Bayes 辨识方法及其 Matlab 仿真，6.3 的内容为卡尔曼滤波与预测，最后一节的内容为模型参考自适应辨识方法。

6.1 Bayes 辨识方法

6.1.1 Bayes 基本原理

Bayes 辨识方法的基本思想是把所要估计的参数看做随机变量，然后设法通过观测与该参数有关联的其他变量，以此来推断这个参数。

设 $\boldsymbol{\mu}$ 是描述某一动态系统的模型，$\boldsymbol{\theta}$ 是模型 $\boldsymbol{\mu}$ 的参数，它会反映在该动态系统的输入输出观测值中。如果系统的输出变量 $z(k)$ 在参数 $\boldsymbol{\theta}$ 及其历史纪录 $\boldsymbol{D}^{(k-1)}$ 条件下的概率密度函数是已知的，记作 $p(z(k)|\boldsymbol{\theta},\boldsymbol{D}^{(k-1)})$，其中 $\boldsymbol{D}^{(k-1)}$ 表示 $(k-1)$ 时刻以前的输入输出数据集合，那么根据 Bayes 的观点，参数 $\boldsymbol{\theta}$ 的估计问题可以看成是把参数 $\boldsymbol{\theta}$ 当作具有某种先验概率密度 $p(\boldsymbol{\theta},\boldsymbol{D}^{(k-1)})$ 的随机变量，如果输入 $u(k)$ 是确定的变量，则利用 Bayes 公式，把参数 $\boldsymbol{\theta}$ 的后验概率密度函数表示成

$$p(\boldsymbol{\theta} \mid \boldsymbol{D}^k) = p(\boldsymbol{\theta} \mid z(k),u(k),\boldsymbol{D}^{(k-1)}) = p(\boldsymbol{\theta} \mid z(k),\boldsymbol{D}^{(k-1)})$$
$$= \frac{p(z(k) \mid \boldsymbol{\theta},\boldsymbol{D}^{(k-1)})p(\boldsymbol{\theta} \mid \boldsymbol{D}^{(k-1)})}{\int_{-\infty}^{\infty} p(z(k) \mid \boldsymbol{\theta},\boldsymbol{D}^{(k-1)})p(\boldsymbol{\theta} \mid \boldsymbol{D}^{(k-1)})\mathrm{d}\boldsymbol{\theta}} \tag{6.1}$$

式中，参数 $\boldsymbol{\theta}$ 的先验概率密度函数 $p(\boldsymbol{\theta}|\boldsymbol{D}^{(k-1)})$ 及数据的条件概率密度函数 $p(z(k)|\boldsymbol{\theta},\boldsymbol{D}^{(k-1)})$ 是已知的；$\boldsymbol{D}^k$ 表示 k 时刻以前的输入输出数据集合，它与 $\boldsymbol{D}^{(k-1)}$ 的关系是

$$\boldsymbol{D}^k = \{z(k),u(k),\boldsymbol{D}^{(k-1)}\} \tag{6.2}$$

而 $u(k)$ 和 $z(k)$ 为系统 k 时刻的输入输出数据。

原则上说，根据式(6.1)可以求得参数 $\boldsymbol{\theta}$ 的后验概率密度函数，但实际上这是困难的，只有在参数 $\boldsymbol{\theta}$ 与数据之间的关系是线性的，噪声又是高斯分析的情况下，才有可能得到式(6.1)的解析解。求得参数 $\boldsymbol{\theta}$ 的后验概率密度函数后，就可利用它进一步求得参数 $\boldsymbol{\theta}$ 的估计值。常用的方法有两种，一种是极大后验参数估计方

法，另一种是条件期望参数估计方法，这两种方法统称为贝叶斯方法。

1. 极大后验参数估计方法

极大后验参数估计方法就是把后验概率密度函数 $p(\boldsymbol{\theta}|\boldsymbol{D}^k)$ 达到极大值作为估计准则[84]。在该准则下求得的参数估计值称作极大后验估计，记作 $\hat{\boldsymbol{\theta}}_{\mathrm{MP}}$。显然，极大后验估计满足方程

$$\left.\frac{\partial p(\boldsymbol{\theta}\mid\boldsymbol{D}^k)}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_{\mathrm{MP}}}=\mathbf{0} \tag{6.3}$$

或

$$\left.\frac{\partial\log p(\boldsymbol{\theta}\mid\boldsymbol{D}^k)}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_{\mathrm{MP}}}=\mathbf{0} \tag{6.4}$$

这意味着在数据$\boldsymbol{D}^k$ 条件下，模型参数 $\boldsymbol{\theta}$ 落在$\hat{\boldsymbol{\theta}}_{\mathrm{MP}}$邻域内的概率比落在其他邻域的概率要大。或者说，$\hat{\boldsymbol{\theta}}_{\mathrm{MP}}$是 $\boldsymbol{\theta}$ 最可能的聚集区域。

如果把式(6.1)代入式(6.4)，并考虑到式(6.1)右边的分母与 $\boldsymbol{\theta}$ 无关，则式(6.4)可写成

$$\left.\frac{\partial\log p(z(k)\mid\boldsymbol{\theta},\boldsymbol{D}^{(k-1)})}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_{\mathrm{MP}}}+\left.\frac{\partial\log p(\boldsymbol{\theta}\mid\boldsymbol{D}^{(k-1)})}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_{\mathrm{MP}}}=\mathbf{0} \tag{6.5}$$

当 $z(k)$是在独立观测条件下的输出样本时，若让式(6.5)左边的第一项为零，则对应的估计值就是极大似然估计。可见，极大后验估计比极大似然估计多考虑了参数 $\boldsymbol{\theta}$ 的先验概率知识。通常情况下，如果参数 $\boldsymbol{\theta}$ 的先验概率密度函数 $p(\boldsymbol{\theta}|\boldsymbol{D}^{(k-1)})$为已知，则极大后验估计将优于极大似然估计。也就是说，极大后验估计的精度将高于极大似然估计。但是，通常由于没有参数 $\boldsymbol{\theta}$ 的先验概率知识，加之参数 $\boldsymbol{\theta}$ 的后验概率密度函数的计算难度较大，因此极大似然估计仍然比极大后验估计用得更普遍。

如果参数 $\boldsymbol{\theta}$ 在取值范围内是均匀分布的，则参数 $\boldsymbol{\theta}$ 的先验概率分布为协方差阵趋于无限大的正态分布，即参数 $\boldsymbol{\theta}$ 的先验概率密度函数可写成

$$p(\boldsymbol{\theta}|\boldsymbol{D}^{(k-1)})=\frac{(2\pi)^{-N/2}}{\sqrt{\det\boldsymbol{P}_{\theta}}}\exp\left\{-\frac{1}{2}(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}})^{\mathrm{T}}\boldsymbol{P}_{\theta}^{-1}(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}})\right\} \tag{6.6}$$

式中，N 表示多元参数 $\boldsymbol{\theta}$ 的维数，即 $N=\dim\boldsymbol{\theta}$；$\bar{\boldsymbol{\theta}}$ 和 $\boldsymbol{P}_{\theta}$ 分别表示参数 $\boldsymbol{\theta}$ 的均值和协方差阵，且 $\boldsymbol{P}_{\theta}^{-1}\rightarrow\mathbf{0}$。那么式(6.5)左边第二项为

$$\left.\frac{\partial\log p(\boldsymbol{\theta}\mid\boldsymbol{D}^{(k-1)})}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_{\mathrm{MP}}}=-\boldsymbol{P}_{\theta}^{-1}(\hat{\boldsymbol{\theta}}_{\mathrm{MP}}-\bar{\boldsymbol{\theta}})\rightarrow\mathbf{0} \tag{6.7}$$

可见，这时极大后验估计就退化成极大似然估计。

上述分析表明，极大后验估计与极大似然估计有着密切的联系，但是它们的基本出发点又是不一样的。极大似然估计立足于直接极大化数据的条件概率密度函数；而极大后验估计则是基于极大化参数 $\boldsymbol{\theta}$ 的后验概率密度函数，它同时考虑了参数 $\boldsymbol{\theta}$ 的先验概率知识。

2. 条件期望参数估计方法

条件期望参数估计方法直接以参数 $\boldsymbol{\theta}$ 的条件数学期望作为参数估计值[82]，即

$$\hat{\boldsymbol{\theta}}(k) = E\{\boldsymbol{\theta} \mid \boldsymbol{D}^k\} = \int_{-\infty}^{\infty} \boldsymbol{\theta} p(\boldsymbol{\theta} \mid \boldsymbol{D}^k) \mathrm{d}\boldsymbol{\theta} \tag{6.8}$$

上式的物理意义是，用随机变量的均值作为它的估计值。

另外，式(6.8)所定义的参数估计值 $\hat{\boldsymbol{\theta}}(k)$ 将等价于极小化参数估计误差的方差的结果，因此条件期望估计有时也称作最小方差估计，即 $\hat{\boldsymbol{\theta}}(k)$ 使得

$$E\{[\boldsymbol{\theta} - \breve{\boldsymbol{\theta}}(k)]^{\mathrm{T}}[\boldsymbol{\theta} - \breve{\boldsymbol{\theta}}(k)] \mid \boldsymbol{D}^k\}\Big|_{\breve{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k)} = \min \tag{6.9}$$

也就是

$$\int_{-\infty}^{\infty} [\boldsymbol{\theta} - \breve{\boldsymbol{\theta}}(k)]^{\mathrm{T}}[\boldsymbol{\theta} - \breve{\boldsymbol{\theta}}(k)] p(\boldsymbol{\theta} \mid \boldsymbol{D}^k) \mathrm{d}\boldsymbol{\theta}\Big|_{\breve{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k)} = \min \tag{6.10}$$

式中，$\breve{\boldsymbol{\theta}}(k)$ 表示参数 $\boldsymbol{\theta}$ 的某种估计量。当 $\breve{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k)$ 时，式(6.10)达到最小值。式(6.10)成立的条件是

$$\frac{\partial \int_{-\infty}^{\infty} [\boldsymbol{\theta} - \breve{\boldsymbol{\theta}}(k)]^{\mathrm{T}}[\boldsymbol{\theta} - \breve{\boldsymbol{\theta}}(k)] p(\boldsymbol{\theta} \mid \boldsymbol{D}^k) \mathrm{d}\boldsymbol{\theta}}{\partial \breve{\boldsymbol{\theta}}(k)}\Bigg|_{\breve{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k)} = 0 \tag{6.11}$$

求导后得

$$\int_{-\infty}^{\infty} [\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(k)] p(\boldsymbol{\theta} \mid \boldsymbol{D}^k) \mathrm{d}\boldsymbol{\theta} = \mathbf{0} \tag{6.12}$$

也就是

$$\int_{-\infty}^{\infty} \hat{\boldsymbol{\theta}}(k) p(\boldsymbol{\theta} \mid \boldsymbol{D}^k) \mathrm{d}\boldsymbol{\theta} = \int_{-\infty}^{\infty} \boldsymbol{\theta} p(\boldsymbol{\theta} \mid \boldsymbol{D}^k) \mathrm{d}\boldsymbol{\theta} \tag{6.13}$$

考虑到 $\int_{-\infty}^{\infty} p(\boldsymbol{\theta} \mid \boldsymbol{D}^k) \mathrm{d}\boldsymbol{\theta} = 1$，得

$$\hat{\boldsymbol{\theta}}(k) = \int_{-\infty}^{\infty} \boldsymbol{\theta} p(\boldsymbol{\theta} \mid \boldsymbol{D}^k) \mathrm{d}\boldsymbol{\theta} = E\{\boldsymbol{\theta} \mid \boldsymbol{D}^k\} \tag{6.14}$$

可见，式(6.8)和(6.9)的定义是等价的。

同时，式(6.8)所定义的参数估计值 $\hat{\boldsymbol{\theta}}(k)$ 又将使参数估计误差的协方差阵取极小值，即

$$E\{[\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)][\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)]^{\mathrm{T}}|\boldsymbol{D}^k\}\Big|_{\breve{\boldsymbol{\theta}}(k)=\hat{\boldsymbol{\theta}}(k)} \tag{6.15}$$

也就是

$$\int_{-\infty}^{\infty}[\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)][\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)]^{\mathrm{T}}p(\boldsymbol{\theta}|\boldsymbol{D}^k)\mathrm{d}\boldsymbol{\theta}\Big|_{\breve{\boldsymbol{\theta}}(k)=\hat{\boldsymbol{\theta}}(k)} \tag{6.16}$$

达到最小值的条件是

$$\frac{\partial\int_{-\infty}^{\infty}[\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)][\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)]^{\mathrm{T}}p(\boldsymbol{\theta}|\boldsymbol{D}^k)\mathrm{d}\boldsymbol{\theta}}{\partial\,\breve{\boldsymbol{\theta}}(k)}\Bigg|_{\breve{\boldsymbol{\theta}}(k)=\hat{\boldsymbol{\theta}}(k)}=0 \tag{6.17}$$

式中

$$\begin{aligned}&\frac{\partial[\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)][\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)]^{\mathrm{T}}}{\partial\,\breve{\boldsymbol{\theta}}(k)}\\&=\left[\frac{\partial[\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)][\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)]^{\mathrm{T}}}{\partial\,\breve{\boldsymbol{\theta}}_1(k)},\cdots,\frac{\partial[\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)][\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)]^{\mathrm{T}}}{\partial\,\breve{\boldsymbol{\theta}}_N(k)}\right]\end{aligned} \tag{6.18}$$

式中第 i 块矩阵为

$$\frac{\partial[\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)][\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}(k)]^{\mathrm{T}}}{\partial\,\breve{\boldsymbol{\theta}}_i(k)}=$$

$$\begin{bmatrix} & -[\boldsymbol{\theta}_1-\breve{\boldsymbol{\theta}}_1(k)] & \\ \mathbf{0} & -[\boldsymbol{\theta}_2-\breve{\boldsymbol{\theta}}_2(k)] & \mathbf{0} \\ & \vdots & \\ -[\boldsymbol{\theta}_1-\breve{\boldsymbol{\theta}}_1(k)]\cdots & -2[\boldsymbol{\theta}_i-\breve{\boldsymbol{\theta}}_i(k)]\cdots & -[\boldsymbol{\theta}_N-\breve{\boldsymbol{\theta}}_N(k)] \\ & \vdots & \\ \mathbf{0} & -[\boldsymbol{\theta}_N-\breve{\boldsymbol{\theta}}_N(k)] & \mathbf{0}\end{bmatrix} \tag{6.19}$$

可见，由式(6.17)必然会导致式(6.13)。也就是说，当 $\hat{\boldsymbol{\theta}}(k)=E\{\boldsymbol{\theta}|\boldsymbol{D}^k\}$ 时，式(6.15)取极小值。

上述分析表明，不管参数 $\boldsymbol{\theta}$ 的后验概率密度函数取什么形式，条件期望参数估计总是无偏一致估计。可是，条件期望参数估计在计算上存在着很大的困难。这是因为计算式(6.8)必须事先求得参数 $\boldsymbol{\theta}$ 的后验概率密度函数，并且式(6.8)的积分运算比较困难。因此，一般情况下，条件期望参数估计在工程上是难以应用的。但是，正如上面提到过的，如果参数 $\boldsymbol{\theta}$ 与输入输出数据之间的关系是线性的，而且数据噪声服从高斯分布，那么式(6.8)将有准确解。下面将主要讨论 Bayes 方法在这种情况下的模型参数辨识问题。

6.1.2 最小二乘模型的 Bayes 参数辨识

考虑模型

$$A(z^{-1})z(k) = B(z^{-1})u(k) + v(k) \tag{6.20}$$

式中，$\{v(k)\}$是均值为零、方差为 σ_v^2 的服从高斯分布的白噪声序列；且

$$\begin{cases} A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a} \\ B(z^{-1}) = b_1 z^{-1} + \cdots + a_{n_b} z^{-n_b} \end{cases} \tag{6.21}$$

模型阶次 n_a 和 n_b 事先给定。

模型(6.20)可以写成最小二乘格式

$$z(k) = \boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta} + v(k) \tag{6.22}$$

式中

$$\begin{cases} \boldsymbol{h}(k) = [-z(k-1),\cdots,-z(k-n_a),u(k-1),\cdots,u(k-n_b)]^{\mathrm{T}} \\ \boldsymbol{\theta} = [a_1,\cdots,a_{n_a},b_1,\cdots,b_{n_b}]^{\mathrm{T}} \end{cases} \tag{6.23}$$

应用 Bayes 方法估计模型(6.22)的参数 $\boldsymbol{\theta}$ 时，首先要把参数 $\boldsymbol{\theta}$ 看做随机变量，然后利用式(6.4)或(6.8)来确定参数 $\boldsymbol{\theta}$ 的估计值。显然，无论利用哪个式子求参数估计值都需要预先确定参数 $\boldsymbol{\theta}$ 的后验概率密度函数 $p(\boldsymbol{\theta}|\boldsymbol{D}^k)$。根据式(6.1)，并利用下面 Bayes 公式

$$p(a,b|c) = p(a|b,c)p(b|c) \tag{6.24a}$$

或进一步写成

$$p(a|b,c) = p(b|a,c)p(a|c)/p(b|c) \tag{6.24b}$$

可得参数 $\boldsymbol{\theta}$ 的后验概率密度函数为

$$p(\boldsymbol{\theta}|\boldsymbol{D}^k) = \frac{p(z(k)|\boldsymbol{\theta},\boldsymbol{D}^{(k-1)})p(\boldsymbol{\theta}|\boldsymbol{D}^{(k-1)})}{p(z(k)|\boldsymbol{D}^{(k-1)})} \tag{6.25}$$

设参数 $\boldsymbol{\theta}$ 在数据 $\boldsymbol{D}^0$ 条件下的先验概率分布是均值为$\hat{\boldsymbol{\theta}}(0)$、协方差阵为 $\boldsymbol{P}(0)$ 的正态分布，即

$$p(\boldsymbol{\theta}|\boldsymbol{D}^0) = \frac{(2\pi)^{-N/2}}{\sqrt{\det\boldsymbol{P}(0)}}\exp\left\{-\frac{1}{2}[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(0)]^{\mathrm{T}}\boldsymbol{P}^{-1}(0)[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(0)]\right\} \tag{6.26}$$

式中，$N=\dim\boldsymbol{\theta}$，则参数 $\boldsymbol{\theta}$ 在数据 $\boldsymbol{D}^k$ 条件下的后验概率分布也是正态分布的，其均值和协方差阵分别记作$\hat{\boldsymbol{\theta}}(k)$和 $\boldsymbol{P}(k)$。于是，参数 $\boldsymbol{\theta}$ 在数据 $\boldsymbol{D}^{(k-1)}$条件下的概率分布可写成

$$p(\boldsymbol{\theta}|\boldsymbol{D}^{(k-1)}) = \frac{(2\pi)^{-N/2}}{\sqrt{\det\boldsymbol{P}(k-1)}}$$

$$\times \exp\left\{-\frac{1}{2}[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]^{\mathrm{T}}\boldsymbol{P}^{-1}(k-1)[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]\right\} \tag{6.27}$$

同时，由于噪声服从正态分布，$v(k)\sim\mathcal{N}(0,\sigma_v^2)$，结合式(6.22)，则有 $z(k)\sim\mathcal{N}(\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta},\sigma_v^2)$，因此数据的条件概率密度函数可写成

$$p(z(k)\mid\boldsymbol{\theta},\boldsymbol{D}^{(k-1)})=\frac{1}{\sqrt{2\pi\sigma_v^2}}\exp\left\{-\frac{1}{2\sigma_v^2}[z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}]^2\right\} \tag{6.28}$$

将式(6.28)和(6.27)代入式(6.25)，得

$$p(\boldsymbol{\theta}\mid\boldsymbol{D}^k)=\mathrm{Norm}\exp\left\{-\frac{1}{2\sigma_v^2}[z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}]^2-\frac{1}{2}[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]^{\mathrm{T}}\right.$$
$$\left.\times\boldsymbol{P}^{-1}(k-1)[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]\right\} \tag{6.29}$$

式中，Norm 与 $\boldsymbol{\theta}$ 无关，且

$$\mathrm{Norm}=\frac{(2\pi)^{-N/2}}{p(z(k)\mid\boldsymbol{D}^{(k-1)})\sqrt{2\pi\sigma_v^2\det\boldsymbol{P}(k-1)}} \tag{6.30}$$

式(6.29)两边取对数后为

$$\log p(\boldsymbol{\theta}\mid\boldsymbol{D}^k)=\mathrm{const}-\frac{1}{2\sigma_v^2}[z(k)-\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}]^2$$
$$-\frac{1}{2}[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)]^{\mathrm{T}}\boldsymbol{P}^{-1}(k-1)[\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}(k-1)] \tag{6.31}$$

将上式右边展开，并整理成二次型

$$\log p(\boldsymbol{\theta}\mid\boldsymbol{D}^k)=\mathrm{const}-\frac{1}{2}\left[\boldsymbol{\theta}-\frac{1}{\sigma_v^2}\boldsymbol{P}(k)\boldsymbol{h}(k)z(k)-\boldsymbol{P}(k)\boldsymbol{P}^{-1}(k-1)\hat{\boldsymbol{\theta}}(k-1)\right]^{\mathrm{T}}$$
$$\times\boldsymbol{P}^{-1}(k)\left[\boldsymbol{\theta}-\frac{1}{\sigma_v^2}\boldsymbol{P}(k)\boldsymbol{h}(k)z(k)\right.$$
$$\left.-\boldsymbol{P}(k)\boldsymbol{P}^{-1}(k-1)\hat{\boldsymbol{\theta}}(k-1)\right] \tag{6.32}$$

式中

$$\boldsymbol{P}^{-1}(k)=\boldsymbol{P}^{-1}(k-1)+\frac{1}{\sigma_v^2}\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k) \tag{6.33}$$

得

$$\boldsymbol{P}(k)\boldsymbol{P}^{-1}(k-1)=\boldsymbol{I}-\frac{1}{\sigma_v^2}\boldsymbol{P}(k)\boldsymbol{h}(k)\boldsymbol{h}^{\mathrm{T}}(k) \tag{6.34}$$

代入式(6.32)，有

$$\log p(\boldsymbol{\theta}\mid\boldsymbol{D}^k)=\mathrm{const}-\frac{1}{2}(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}})^{\mathrm{T}}\boldsymbol{P}^{-1}(k)(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}}) \tag{6.35}$$

式中

$$\bar{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(k-1) + \frac{1}{\sigma_v^2}\boldsymbol{P}(k)\boldsymbol{h}(k)[z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1)] \tag{6.36}$$

根据式(6.4)，当$\hat{\boldsymbol{\theta}}(k)=\bar{\boldsymbol{\theta}}$时，$\log p(\boldsymbol{\theta}|\boldsymbol{D}^k)$达到最大值。因此$\hat{\boldsymbol{\theta}}(k)=\bar{\boldsymbol{\theta}}$是参数$\boldsymbol{\theta}$在$k$时刻的极大后验估计。

同时，式(6.35)意味着参数$\boldsymbol{\theta}$的后验概率密度函数为

$$p(\boldsymbol{\theta}|\boldsymbol{D}^k) = \mathrm{Norm}\exp\left\{-\frac{1}{2}(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}})^{\mathrm{T}}\boldsymbol{P}^{-1}(k)(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}})\right\} \tag{6.37}$$

根据式(6.8)，有

$$\hat{\boldsymbol{\theta}}(k) = E\{\boldsymbol{\theta}|\boldsymbol{D}^k\} = \bar{\boldsymbol{\theta}} \tag{6.38}$$

可见，模型式(6.22)的极大后验参数估计和条件期望参数估计两者的结果是一致的。但是，这并不能说两种参数估计方法对所有问题的估计结果都是一致的。一般说来，当k比较小时，这两种方法的估计结果是不同的；当k比较大时，它们就没有什么差别了，两者的估计结果将趋于一致。

如果对式(6.33)使用一次矩阵反演公式，并令

$$\boldsymbol{K}(k) = \frac{1}{\sigma_v^2}\boldsymbol{P}(k)\boldsymbol{h}(k) \tag{6.39}$$

则类似于最小二乘递推算法的推导，可求得 Bayes 方法的参数递推估计算法为

$$\begin{cases}\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{K}(k)[z(k) - \boldsymbol{h}^{\mathrm{T}}(k)\hat{\boldsymbol{\theta}}(k-1)] \\ \boldsymbol{K}(k) = \boldsymbol{P}(k-1)\boldsymbol{h}(k)[\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}(k) + \sigma_v^2]^{-1} \\ \boldsymbol{P}(k) = [\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{h}^{\mathrm{T}}(k)]\boldsymbol{P}(k-1)\end{cases} \tag{6.40}$$

显然，在假设正态分布的前提下，Bayes 估计和 Markov 估计是一致的，它相当于加权最小二乘法中的加权因子取$\Lambda(k)=1/\sigma_v^2$。

6.2 Bayes 辨识的 MATLAB 仿真

例 6.1 考虑图 6.1 所示的仿真对象，图中，$v(k)$是均值为零、方差为σ_v^2的服从正态分布的不相关随机噪声，且

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})}, \quad N(z^{-1}) = \frac{D(z^{-1})}{C(z^{-1})}$$

图 6.1 Bayes 辨识结构图

$$
\begin{cases}
A(z^{-1}) = 1 - 1.5a_1 z^{-1} + 0.7z^{-2} = C(z^{-1}) \\
B(z^{-1}) = 1.0z^{-1} + 0.5z^{-2} \\
D(z^{-1}) = 1 - z^{-1} + 0.2z^{-2}
\end{cases}
$$

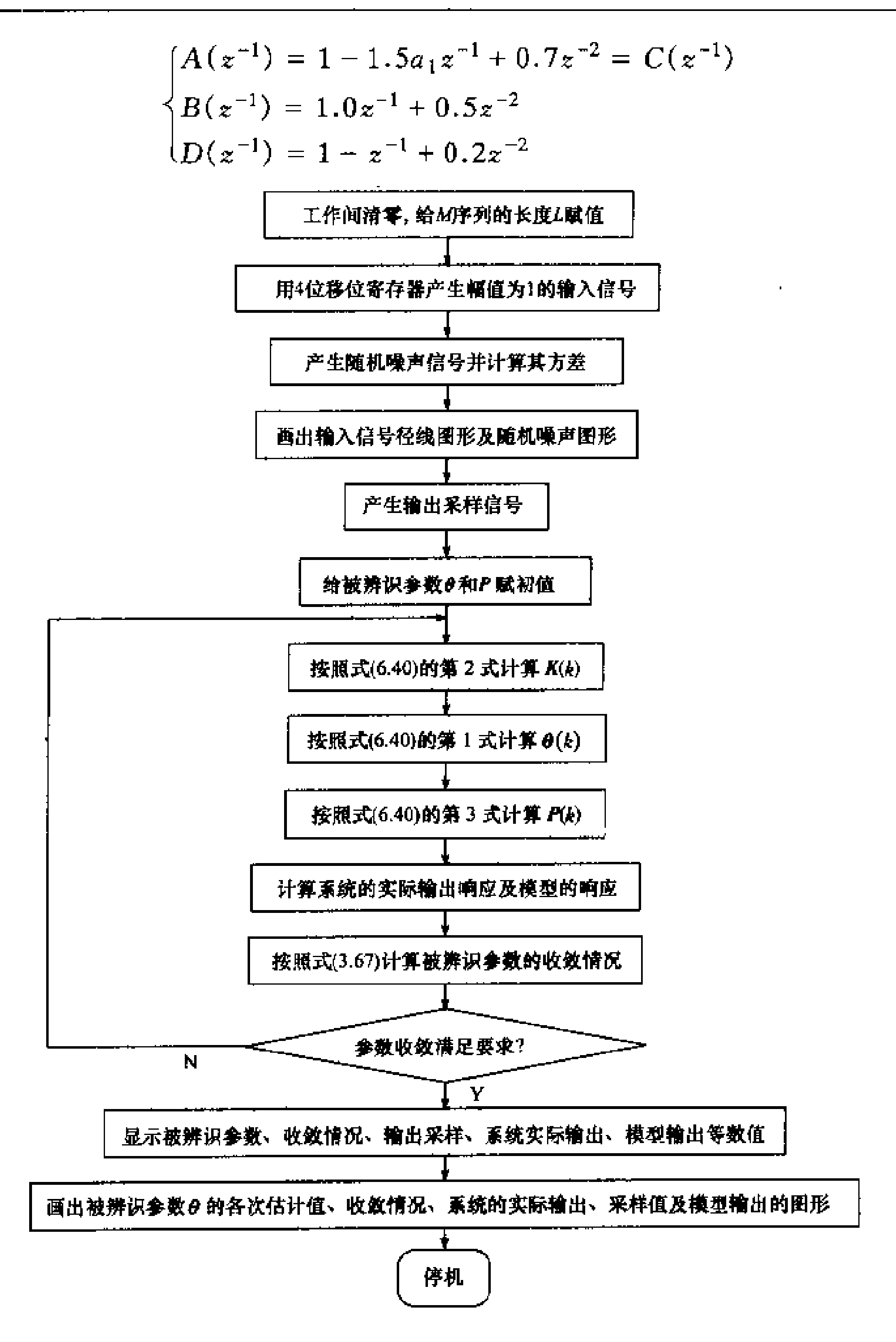


图 6.2　Bayes 辨识的 MATLAB 6.0 程序流程图

模型结构选用如下形式

$$z(k)+a_1z(k-1)+a_2z(k-2)$$
$$=b_1u(k-1)+b_2u(k-2)+v(k)+d_1v(k-1)+d_2(v(k-2))$$

Bayes 辨识算法程序流程如图 6.2 所示，MATLAB 6.0 程序如下。

```
% Bayes 辨识程序,在光盘中的文件名:FLch6BAeg1.m。
clear
L=60; %四位移位寄存器产生的 M 序列的长度
y1=1;y2=1;y3=1;y4=0; %四个移位寄存器的输出初始值
for i=1:L;
    x1=xor(y3,y4); %第一个移位寄存器的输入信号
    x2=y1; %第二个移位寄存器的输入信号
    x3=y2; %第三个移位寄存器的输入信号
    x4=y3; %第四个移位寄存器的输入信号
    y(i)=y4; %第四个移位寄存器的输出信号
    if y(i)>0.5,u(i)=-1; %M 序列的值为"1"时,辨识的输入信号取"-1"
    else u(i)=1; %M 序列的值为"0"时,辨识的输入信号取"1"
    end
    y1=x1;y2=x2;y3=x3;y4=x4; %为下一次的输入信号作准备
end
figure(1);subplot(2,1,1);stem(u),grid on %画第一个图形的第一个子图——M 序列输入信号
v=randn(1,60);subplot(2,1,2); %产生一组长度为 60 的正态分布随机噪声
plot(v),grid on; %画第一个图形窗口的第二个子图——随机噪声信号
R=corrcoef(u,v); %计算输入信号与随机噪声信号的相关系数
r=R(1,2) %取出并显示互相关系数
rv=std(v)*std(v) %计算并显示随机噪声的方差
u,v %显示输入信号和噪声信号
z(2)=0;z(1)=0;zs(2)=0;zs(1)=0; %输出采样、系统实际输出、模型输出赋初值
zm(2)=0;zm(1)=0; %模型输出赋初值
%Bayes 辨识
c0=[0.001 0.001 0.001 0.001 0.001 0.001 0.001]'; %直接给出被辨识参数的初始值,即一个充分小的
%实向量
p0=10^6*eye(7,7); %直接给出初始状态 P0,即一个充分大的实数单位矩阵
E=0.000000005; %取相对误差 E<=0.000000005 为停机准则
c=[c0,zeros(7,14)]; %被辨识参数矩阵的初始值及大小
e=zeros(7,15); %相对误差的初始值及大小
for k=3:20; %开始求 K
    z(k)=-1.5*z(k-1)+0.7*z(k-2)+u(k-1)+0.5*u(k-2)+v(k)-v(k-1)+0.2*v(k-2);
    %系统在 M 序列输入下的输出采样信号
    h1=[-z(k-1),-z(k-2),u(k-1),u(k-2),v(k),v(k-1),v(k-2)]'; %为求 K(k)作准备
    x=h1'*p0*h1+rv;
    x1=inv(x);
```

```
    k1 = p0 * h1 * x1; %K
    d1 = z(k) - h1' * c0; %开始求被辨识参数 c
    c1 = c0 + k1 * d1; %辨识参数 c
    zs(k) = -1.5 * z(k-1) + 0.7 * z(k-2) + u(k-1) + 0.5 * u(k-2); %系统在 M 序列输入下的输出响应
    zm(k) = [-z(k-1), -z(k-2), u(k-1), u(k-2)] * [c1(1); c1(2); c1(3); c1(4)]; %模型的输出响应
    e1 = c1 - c0;
    e2 = e1./c0; %求参数的相对变化
    e(:,k) = e2;
    c0 = c1; %给下一次用
    c(:,k) = c1; %把辨识参数 c 列向量加入辨识参数矩阵
    p1 = p0 - k1 * k1' * [h1' * p0 * h1 + 1]; %find p(k)
    p0 = p1; %给下次用
    if e2 <= E break; %若收敛情况满足要求,终止计算
    end
end
c,e,z,zs,zm %显示被辨识参数、收敛情况、输出采样值、实际输出值、模型输出值
%分离赋值
a1 = c(1,:); a2 = c(2,:); b1 = c(3,:); b2 = c(4,:); %分离出 a1、a2、b1、b2
d1 = c(5,:); d2 = c(6,:); d3 = c(7,:); %分离出 d1、d2、d3
ea1 = e(1,:); ea2 = e(2,:); eb1 = e(3,:); eb2 = e(4,:); %分离出 a1、a2、b1、b2 的收敛情况
ed1 = e(5,:); ed2 = e(6,:); ed3 = e(7,:); %分离出 d1、d2、d3 的收敛情况
figure(2); %画第二个图形
i = 1:20;
plot(i,a1,'r',i,a2,'r:',i,b1,'b',i,b2,'b:',i,d1,'g',i,d2,'g:',i,d3,'g+') %画出被辨识参数的各次估计值
title('Parameter Identification with Recursive Least Squares Method') %标题
figure(3); %画出第三个图形
i = 1:20;
plot(i,ea1,'r',i,ea2,'r:',i,eb1,'b',i,eb2,'b:',i,ed1,'g',i,ed2,'g:',i,ed2,'r+') %画出各个参数收敛情况
title('Identification Error') %标题
    %Response %响应
figure(4); subplot(4,1,1); %画出第四个图形中的第一个子图
i = 1:20; plot(i,zs(i),'r') %画出被辨识系统在没有噪声情况下的实际输出响应
subplot(4,1,2); %画出第四个图形中的第二个子图
i = 1:20; plot(i,z(i),'g') %画出被辨识系统输出的采样值
subplot(4,1,3); %画出第四个图形中的第三个子图
i = 1:20; plot(i,zm(i),'b') %画出模型含有噪声的输出响应
subplot(4,1,4); %画出第四个图形中的第四个子图
i = 1:20; plot(i,zmy(i),'b') %画出模型去除噪声后的输出响应
```

程序执行结果如下:

```
u =
[1,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,1,-1,1,-1,-1,1,-1,-1,1,1,-1,1,1,-1,-1,1,-1,1, 1, 1,
- ,-1,1,1,1,-1,1,1,-1,-1,1,-1,1,-1,-1, -1,-1,1,1,1,-1,1,1,-1,-1,1, 1]
```

```
v=
[-0.4326,-1.6656,0.1253,0.2877,-1.1465,1.1909,1.1892,-0.0376,0.3273,0.1746,-0.1867,0.7258,-0.5883,2.1832,
-0.1364,0.1139,1.0668,0.0593,-0.0956,-0.8323,0.2944,-1.3362,0.7143,1.6236,-0.6918,0.8580,1.2540,-1.5937,
-1.4410,0.5711,-0.3999,0.6900,0.8156,0.7119,1.2902,0.6686,1.1908,-1.2025,-0.0198,-0.1567,-1.6041,0.2573,
-1.0565,1.4151,-0.8051,0.5287,0.2193,-0.9219,-2.1707,-0.0592,-1.0106,0.6145,0.5077,1.6924,0.5913,-0.6436,
0.3803,-1.0091,-0.0195,-0.0482]
rv=
0.8991
r=
0.0591
c=
0.0010 0   0.0010  -0.0004  -0.1558  -0.9692  -1.2967  -1.3339  -1.5000  -1.5000  -1.5000  -1.5000
0.0010 0   0.0010   0.0010   0.9464   0.9030   0.3195   0.4045   0.7000   0.7000   0.7000   0.7000
0.0010 0   0.2414  -0.2425   0.4557   0.6715   0.7941   0.6938   1.0000   1.0000   1.0000   1.0000
0.0010 0   0.2434   0.2421   0.6788   1.1565   1.0305   1.0194   0.5000   0.5000   0.5000   0.5000
0.0010 0   0.0314   0.0317   0.9779   0.5510   1.1953   1.2261   1.0000   1.0000   1.0000   1.0000
0.0010 0   0.4027  -0.4024  -0.3832  -0.2854  -0.4734  -0.4439  -1.0000  -1.0000  -1.0000  -1.0000
0.0010 0  -0.1038  -0.1058  -0.5099  -0.4045  -0.0686   0.0328   0.2000   0.2000   0.2000   0.2000
e=
0 0     0        -1.4450  349.1896   5.2201   0.3379   0.0287   0.1245  -0.0000   0.0000   0.0000
0 0     0         0       945.4254  -0.0458  -0.6462   0.2662   0.7305  -0.0000   0.0000  -0.0000
0 0  -242.3907    0.0044   -2.8795   0.4734   0.1827  -0.1263   0.4413   0.0000   0.0000  -0.0000
0 0   242.3907   -0.0055    1.8042   0.7038  -0.1089  -0.0109  -0.5095  -0.0000  -0.0000   0.0000
0 0    30.3794    0.0105   29.8394  -0.4365   1.1693   0.0258  -0.1844  -0.0000  -0.0000   0.0000
0 0  -403.7221   -0.0009   -0.0477  -0.2552   0.6588  -0.0623   1.2527   0.0000   0.0000  -0.0000
0 0  -104.8497    0.0187    3.8203  -0.2068  -0.8305  -1.4781   5.0998  -0.0000   0.0000   0.0000
```

Bayes 辨识结果见表 6.1。仿真结果表明，Bayes 最小二乘辨识，递推到第 9 步时，参数辨识的结果达到稳定状态，即 $a_1=-1.5000, a_2=0.7000, b_1=1.0000, b_2=0.5000, d_1=1.0000, d_2=-1.0000, d_3=0.2000$。此时，辨识参数的相对变化量 $E \leqslant 0.000\,000\,005$。可见，此例的 Bayes 最小二乘递推算法的辨识结果与增广最小二乘递推算法的辨识结果是一致的。

表 6.1　Bayes 辨识结果

参　数	a_1	a_2	b_1	b_2	d_1	d_2	d_3
真　值	-1.5	0.7	1.0	0.5	1.0	-1.0	0.2
估计值	-1.5	0.7	1.0	0.5	1.0	-1.0	0.2

程序中，预先把参数的估计值矩阵、参数的收敛状况矩阵及系统和模型的响应矩阵的初始值均设定为零，所以，当参数的收敛状况满足要求时，程序就停止运行。在图 6.3 中可以看到，各项计算数值和输出数值均变为零的时刻，即为程序停止运行的时刻。随着 E 的取值的减小，程序停止运行的时刻将向后推移。

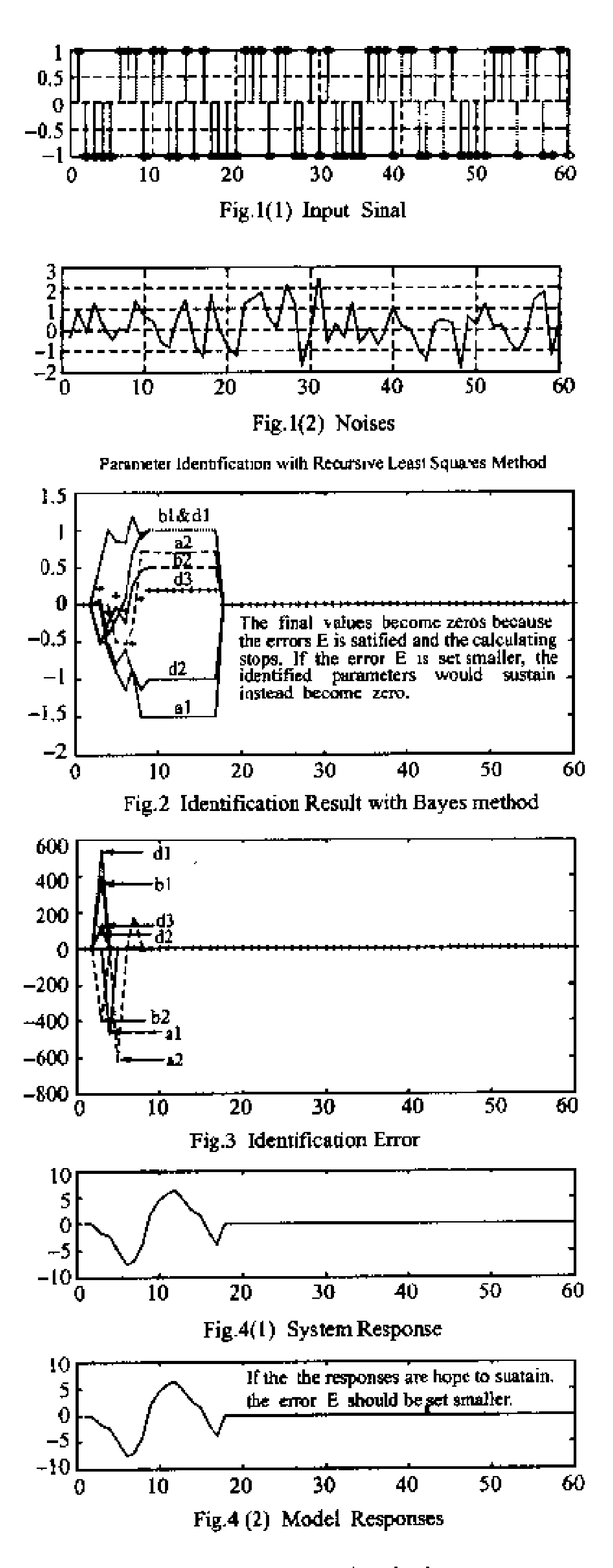


图 6.3　Bayes 辨识仿真

需要强调指出，加在辨识系统上的噪声是随机噪声，即每次辨识过程中的噪声是惟一的，所以每一次程序运行的中间过程是不同的，但参数辨识的最终结果是一致的。

6.3 Kalman 滤波

Kalman 滤波是现代控制理论的一个重要支柱。它有着一个递推的表述形式，适合于实时预测与滤波，且易于在数字计算机上实现，因此在实际问题中得到了广泛的应用[44,48,79]。本节内容为卡尔曼理论最初讨论的离散线性时变系统状态最小方差滤波与预测问题。

6.3.1 预测、滤波与平滑

预测、滤波与平滑，它们的总称就是估计[49]。下面具体解释预测、滤波与平滑的概念。

设 $\boldsymbol{x}(k)$为待估计的矢量，$\boldsymbol{y}_j=[\boldsymbol{y}(1)^{\mathrm{T}},\boldsymbol{y}(2)^{\mathrm{T}},\cdots,\boldsymbol{y}(j)^{\mathrm{T}}]^{\mathrm{T}}$ 为对 $\boldsymbol{x}(k)$的实测值。若能根据 $\boldsymbol{y}_j$决定一个$\boldsymbol{x}(k)$的近似值

$$\hat{\boldsymbol{x}}(k \mid j) = f(\boldsymbol{y}_j) \tag{6.41}$$

则$\hat{\boldsymbol{x}}(k|j)$称为根据 $\boldsymbol{y}_j$决定的$\boldsymbol{x}(k)$的估计。而

$$\tilde{\boldsymbol{x}}(k \mid j) = \boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k \mid j) \tag{6.42}$$

称为$\tilde{\boldsymbol{x}}(k)$的估计误差。若

$$\hat{\boldsymbol{x}}(k \mid j) = \boldsymbol{A}(\boldsymbol{y}_j) + \boldsymbol{b} \tag{6.43}$$

式中，$\boldsymbol{A}$ 为矩阵，$\boldsymbol{b}$ 为矢量。则这种特殊的估计称为线性估计。$k>j$ 的估计称为预测或外推；$k=j$ 的估计称为滤波；$k<j$ 的估计称为平滑或内插。

若误差的均值

$$E[\tilde{\boldsymbol{x}}(k \mid j)] = E[\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k \mid j)] = 0 \tag{6.44}$$

则称$\hat{\boldsymbol{x}}(k|j)$为 $\boldsymbol{x}(k)$之无偏估计。对无偏估计而言，估计$\hat{\boldsymbol{x}}(k|j)$以相等的概率分布在 $\boldsymbol{x}(k)$的两边。

设常矢量 $\boldsymbol{a}$ 和$\boldsymbol{x}(k)$同维数，则 $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}(k)$表示 $\boldsymbol{a}$ 和$\boldsymbol{x}(k)$的某个分量或某些分量的线性组合。若估计$\hat{\boldsymbol{x}}(k|j)$使得准则函数

$$J = E[\boldsymbol{a}^{\mathrm{T}}\tilde{\boldsymbol{x}}(k \mid j)][\boldsymbol{a}^{\mathrm{T}}\tilde{\boldsymbol{x}}(k \mid j)]^{\mathrm{T}} = \min \tag{6.45}$$

则，$\hat{\boldsymbol{x}}(k|j)$称为 $\boldsymbol{x}(k)$的最小方差估计。根据参考文献[49]可知，当用$E[\boldsymbol{x}(k)|\boldsymbol{y}_j]$作为 $\boldsymbol{x}(k)$的估计值时，它既是无偏估计，又是最小方差估计。

6.3.2 高斯变量估计

若待估计矢量 $\boldsymbol{x}(k)$ 与实测值 $\boldsymbol{y}_j$ 的联合分布是高斯型的，那么 $\boldsymbol{x}(k)$ 的最小方差估计将有一个明显的表达式。下面就来决定这一表达式。简记 $\boldsymbol{x}(k)=\boldsymbol{x}$，$\boldsymbol{y}_j=\boldsymbol{y}$，设 $(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$ 服从高斯分布，且

$$E\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{pmatrix}=\begin{pmatrix}\bar{\boldsymbol{x}}\\ \bar{\boldsymbol{y}}\end{pmatrix}=\boldsymbol{m} \tag{6.46}$$

$$\begin{aligned}&E\begin{pmatrix}\boldsymbol{x}-\bar{\boldsymbol{x}}\\ \boldsymbol{y}-\bar{\boldsymbol{y}}\end{pmatrix}((\boldsymbol{x}-\bar{\boldsymbol{x}})^{\mathrm{T}},(\boldsymbol{y}-\bar{\boldsymbol{y}})^{\mathrm{T}})\\ &=\begin{pmatrix}\mathrm{var}(\boldsymbol{x}) & \mathrm{cov}(\boldsymbol{x},\boldsymbol{y})\\ \mathrm{cov}(\boldsymbol{y},\boldsymbol{x}) & \mathrm{var}(\boldsymbol{y})\end{pmatrix}\\ &=\begin{pmatrix}\boldsymbol{P}_x & \boldsymbol{R}_{xy}\\ \boldsymbol{R}_{yx} & \boldsymbol{P}_y\end{pmatrix}=\boldsymbol{R}\end{aligned} \tag{6.47}$$

则

$$E(\boldsymbol{x}\mid\boldsymbol{y})=\bar{\boldsymbol{x}}+\mathrm{cov}(\boldsymbol{x},\boldsymbol{y})\mathrm{var}^{-1}(\boldsymbol{y})(\boldsymbol{y}-\bar{\boldsymbol{y}}) \tag{6.48}$$

而估计误差的方差

$$\begin{aligned}\mathrm{var}(\tilde{\boldsymbol{x}})&=E[\boldsymbol{x}-E(\boldsymbol{x}\mid\boldsymbol{y})][\boldsymbol{x}-E(\boldsymbol{x}\mid\boldsymbol{y})]^{\mathrm{T}}\\ &=\mathrm{var}(\boldsymbol{x})-\mathrm{cov}(\boldsymbol{x},\boldsymbol{y})\mathrm{var}^{-1}(\boldsymbol{y})\cdot\mathrm{cov}(\boldsymbol{y},\boldsymbol{x})\end{aligned} \tag{6.49}$$

很显然，式(6.48)给出了 $\boldsymbol{x}(k)$ 的最小方差估计，而式(6.49)给出了相应误差的方差。由于式(6.46)中的 $\boldsymbol{m}$ 与式(6.47)中的 $\boldsymbol{R}$ 都不依赖于 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 的取值，所以，高斯变量的最小方差估计是一种线性估计。下面给出式(6.48)和(6.49)的证明。

做矢量

$$\boldsymbol{v}=\boldsymbol{x}-\bar{\boldsymbol{x}}-\boldsymbol{R}_{xy}\boldsymbol{P}_y^{-1}(\boldsymbol{y}-\bar{\boldsymbol{y}}) \tag{6.50}$$

则

$$\bar{\boldsymbol{v}}=\bar{\boldsymbol{x}}-\bar{\boldsymbol{x}}-\boldsymbol{R}_{xy}\boldsymbol{P}_y^{-1}(\bar{\boldsymbol{y}}-\bar{\boldsymbol{y}})=\boldsymbol{0} \tag{6.51}$$

故协方差

$$\begin{aligned}\mathrm{cov}(\boldsymbol{v},\boldsymbol{y})&=E[\boldsymbol{v}(\boldsymbol{y}-\bar{\boldsymbol{y}})^{\mathrm{T}}]\\ &=E(\boldsymbol{x}-\bar{\boldsymbol{x}})(\boldsymbol{y}-\bar{\boldsymbol{y}})^{\mathrm{T}}\\ &\quad-\boldsymbol{R}_{xy}\boldsymbol{P}_y^{-1}E(\boldsymbol{y}-\bar{\boldsymbol{y}})(\boldsymbol{y}-\bar{\boldsymbol{y}})^{\mathrm{T}}\\ &=\boldsymbol{R}_{xy}-\boldsymbol{R}_{xy}\boldsymbol{P}_y^{-1}\boldsymbol{R}_y=\boldsymbol{0}\end{aligned} \tag{6.52}$$

即，$\boldsymbol{v}$ 与 $\boldsymbol{y}$ 不相关。又因 $\boldsymbol{v}$ 是 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 的线性函数，$\boldsymbol{v}$ 也应是高斯变量，故 $\boldsymbol{v}$ 与 $\boldsymbol{y}$ 互相独立。而 $\boldsymbol{v}$ 的方差

$$\begin{aligned}\mathrm{var}(\boldsymbol{v})&=E(\boldsymbol{v}-\bar{\boldsymbol{v}})(\boldsymbol{v}-\bar{\boldsymbol{v}})^{\mathrm{T}}\\ &=E(\boldsymbol{x}-\bar{\boldsymbol{x}})(\boldsymbol{x}-\bar{\boldsymbol{x}})^{\mathrm{T}}-E(\boldsymbol{x}-\bar{\boldsymbol{x}})(\boldsymbol{y}-\bar{\boldsymbol{y}})^{\mathrm{T}}\boldsymbol{P}_y^{-1}\boldsymbol{R}_{yx}\end{aligned}$$

$$
\begin{aligned}
&- \boldsymbol{R}_x\boldsymbol{P}_y^{-1}E(\boldsymbol{y}-\bar{\boldsymbol{y}})(\boldsymbol{x}-\bar{\boldsymbol{x}})^{\mathrm{T}} \\
&+ \boldsymbol{R}_{xy}\boldsymbol{P}_y^{-1}E(\boldsymbol{y}-\bar{\boldsymbol{y}})(\boldsymbol{y}-\bar{\boldsymbol{y}})^{\mathrm{T}}\boldsymbol{P}_y^{-1}\boldsymbol{R}_{yx} = \boldsymbol{P}_x - \boldsymbol{P}_{xy}\boldsymbol{P}_y^{-1}\boldsymbol{R}_{yx} \\
= &\operatorname{var}(\boldsymbol{x}) - \operatorname{cov}(\boldsymbol{x},\boldsymbol{y})\operatorname{var}^{-1}(\boldsymbol{y})\operatorname{cov}(\boldsymbol{y},\boldsymbol{x})
\end{aligned} \tag{6.53}
$$

显然

$$
\begin{pmatrix} \boldsymbol{x}-\bar{\boldsymbol{x}} \\ \boldsymbol{y}-\bar{\boldsymbol{y}} \end{pmatrix} = \begin{bmatrix} \boldsymbol{I} & R_{xy}P_y^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{y}-\bar{\boldsymbol{y}} \end{pmatrix} \tag{6.54}
$$

设 $\boldsymbol{x}$ 与 $\boldsymbol{v}$ 是 n 维随机向量，当已知 $(\boldsymbol{v}^{\mathrm{T}},\boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$ 的密度函数 $f(\boldsymbol{v})f(\boldsymbol{y})$ 时，$(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$ 的密度函数

$$
f(\boldsymbol{x},\boldsymbol{y}) = f(\boldsymbol{v})f(\boldsymbol{y})\left|\frac{\partial(\boldsymbol{x},\boldsymbol{y})}{\partial(\boldsymbol{v},\boldsymbol{y})}\right| \tag{6.55}
$$

而式中

$$
\frac{\partial(\boldsymbol{x},\boldsymbol{y})}{\partial(\boldsymbol{v},\boldsymbol{y})} = \det\begin{bmatrix} \dfrac{\partial \boldsymbol{x}}{\partial \boldsymbol{v}} & \dfrac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}} \\ \dfrac{\partial \boldsymbol{y}}{\partial \boldsymbol{v}} & \dfrac{\partial \boldsymbol{y}}{\partial \boldsymbol{y}} \end{bmatrix} = \det\begin{bmatrix} \boldsymbol{I} & \boldsymbol{R}_{xy}\boldsymbol{P}_y^{-1} \\ \boldsymbol{0} & I \end{bmatrix} = 1 \tag{6.56}
$$

是变量 $(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$ 相对变量的雅可比行列式。故有

$$
\begin{aligned}
f(\boldsymbol{x}\mid \boldsymbol{y}) &= \frac{f(\boldsymbol{x},\boldsymbol{y})}{f(\boldsymbol{y})} = f(\boldsymbol{v}) = \frac{1}{(2\pi)^{n/2}(\det\boldsymbol{P}_v)^{1/2}}\exp(-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{P}_v^{-1}\boldsymbol{v}) \\
&= \frac{1}{(2\pi)^{n/2}(\det\boldsymbol{P}_v)^{1/2}} \\
&\quad\times\exp\left\{-\frac{1}{2}[\boldsymbol{x}-\bar{\boldsymbol{x}}-\boldsymbol{R}_{xy}\boldsymbol{P}_y^{-1}(\boldsymbol{y}-\boldsymbol{y})]^{\mathrm{T}}\boldsymbol{P}_v^{-1}[\boldsymbol{x}-\bar{\boldsymbol{x}}-\boldsymbol{R}_{xy}\boldsymbol{P}_y^{-1}(\boldsymbol{y}-\bar{\boldsymbol{y}})]\right\}
\end{aligned} \tag{6.57}
$$

很显然，在已知 $\boldsymbol{y}$ 的条件下，$\boldsymbol{x}$ 依然服从高斯分布，且其条件均值

$$
E(\boldsymbol{x}\mid \boldsymbol{y})\boldsymbol{v} = \hat{\boldsymbol{x}} = \bar{\boldsymbol{x}} + \boldsymbol{R}_{xy}\boldsymbol{P}_y^{-1}(\boldsymbol{y}-\bar{\boldsymbol{y}}) \tag{6.58}
$$

此式表明，式(6.50)所示之 $\boldsymbol{v}$ 即 $\boldsymbol{x}$ 的最小方差估计的误差，也就是说

$$
\boldsymbol{v} = \tilde{\boldsymbol{x}} \tag{6.59}
$$

而式(6.57)的第二个等号的右边又给出了 $\boldsymbol{v}$ 的密度函数，故有

$$
\operatorname{var}(\tilde{\boldsymbol{x}}) = \operatorname{var}(\boldsymbol{v}) = \operatorname{var}(\boldsymbol{x}) - \operatorname{cov}(\boldsymbol{x},\boldsymbol{y})\operatorname{var}^{-1}(\boldsymbol{y})\operatorname{cov}(\boldsymbol{y},\boldsymbol{x}) \tag{6.60}
$$

于是，式(6.48)和(6.49)得证。

若对 $\boldsymbol{x}$ 有两组独立的测量 $\boldsymbol{y}$ 与 $\boldsymbol{z}$，且 $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ 的联合分布是高斯型的，那么可将向量 $(\boldsymbol{y}^{\mathrm{T}},\boldsymbol{z}^{\mathrm{T}})^{\mathrm{T}}$ 看做是测量值，而有 $\boldsymbol{x}$ 的最小方差估计

$$
\hat{\boldsymbol{x}} = E(\boldsymbol{x}\mid \boldsymbol{y},\boldsymbol{z}) = E(\boldsymbol{x}\mid \boldsymbol{y}) + E(\boldsymbol{x}\mid \boldsymbol{z}) - \bar{\boldsymbol{x}} \tag{6.61}
$$

现将上式的第二个等号证明如下，令

$$u = (y^{\mathrm{T}}, z^{\mathrm{T}})^{\mathrm{T}} \tag{6.62}$$

则

$$\mathrm{var}(u) = \begin{bmatrix} P_u & 0 \\ 0 & P_z \end{bmatrix} \tag{6.63}$$

而

$$\begin{aligned} \mathrm{cov}(x, u) &= E(x - \bar{x})(u - \bar{u})^{\mathrm{T}} \\ &= E(x - \bar{x})\begin{bmatrix} (y - \bar{y})^{\mathrm{T}} \\ (z - \bar{z})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = (R_{xy}, R_{xz}) \end{aligned} \tag{6.64}$$

由式(6.48)得

$$\begin{aligned} E(x \mid y, z) &= E(x \mid u) \\ &= \bar{x} + \mathrm{cov}(x, u)\mathrm{var}^{-1}(u)(u - \bar{u})^{\mathrm{T}} \\ &= \bar{x} + R_{xy}P_y^{-1}(y - \bar{y}) + R_{xz}P_z^{-1}(z - \bar{z}) \\ &= E(x \mid y) + E(x \mid z) - \bar{x} \end{aligned} \tag{6.65}$$

于是,式(6.61)得证。

若对 x 有两组非独立的测量 y 与 z,且 (x, y, z) 的联合分布仍是高斯型的,那么 x 的最小方差估计

$$\hat{x} = E(x \mid y, z) = E(x \mid y, \tilde{z}) \tag{6.66}$$

式中,$\tilde{z}$ 是以 y 为测量值对 z 的最小方差估计 $\hat{z}$ 的估计误差,即

$$\tilde{z} = z - E(z \mid y) = z - \bar{z} - R_{xy}P_y^{-1}(y - \bar{y}) \tag{6.67}$$

且

$$E(\tilde{z}) = 0 \tag{6.68}$$

$$\mathrm{cov}(\tilde{z}, y) = 0 \tag{6.69}$$

根据式(6.61),同时考虑到

$$E(x \mid y, z) = E(x \mid y, \tilde{z}) \tag{6.70}$$

式(6.66)还可以写为

$$\hat{x} = E(x \mid y) + E(x \mid \tilde{z}) - \bar{x} \tag{6.71}$$

式(6.66)的重要性在于,它将相关测量下的估计问题转化为等效的互相独立的测量下的估计问题。这样做的结果可能导致计算工作得到某种简化。

6.3.3 Kalman 滤波与预测

对于线性随机系统的自适应滤波问题而言,自适应滤波器就是一个具有学习功能的 Kalman 滤波器。或者说,自适应滤波器可以根据所得测量数据做出必要

的决策，以调整滤波程序，使之适应系统或环境的变化，从而得到接近最优的状态估计。图 6.4 给出了这种滤波器的结构。

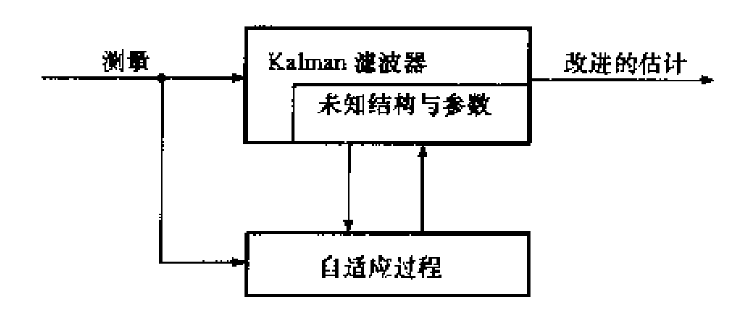


图 6.4 自适应滤波的结构示意图

Kalman 滤波是把不可观测的待估计的状态变量 x 看做随机变量，这些状态变量与可观测的输入输出变量是密切相关的，Kalman 滤波正是基于这些可观测的输入输出变量去推断那些不可观测的状态变量。

现假设系统的状态方程与测量方程分别是

$$\begin{cases} x(k+1) = \phi(k+1,k)x(k) + \eta(k)\omega(k) \\ z(k) = \vartheta(k)x(k) + v(k) \end{cases} \tag{6.72}$$

式中，$\{\omega(k), k\in T\}$，$\{v(k), k\in T\}$ 均为高斯序列，T 是 k 的集合；$\{x(0)\}$ 为高斯变量。且

$$\begin{cases} \bar{\omega}(k) = \bar{v}(k) = 0, \quad \bar{x}(0) = m \\ \mathrm{cov}[\omega(k),\omega(j)] = R_1(k)\delta_{kj} \\ \mathrm{cov}[v(k),v(j)] = R_2(k)\delta_{kj} \\ \mathrm{var}[x(0)] = Px(0) = P_0 \\ \mathrm{cov}[\omega(k),v(j)] = \mathrm{cov}[\omega(k),x(0)] = \mathrm{cov}[v(k),x(0)] = 0 \end{cases} \tag{6.73}$$

式中，$\boldsymbol{R}_1(k)$、$\boldsymbol{P}_0$ 为非负定阵，$\boldsymbol{R}_2(k)$ 为正定阵，以上诸条件在 $k,j\in T$ 时都存在。

1．Kalman 预测公式

设 $k>j$，由式(6.72)知，k 与 j 两瞬时的状态满足方程

$$x(k) = \phi(k,j)x(j) + \sum_{i=j+1}^{k}\phi(k,i)\eta(i-1)\omega(i-1) \tag{6.74}$$

若已知直到 j 瞬时的实测的输出值

$$y_j^{\mathrm{T}} = (y^{\mathrm{T}}(1), y^{\mathrm{T}}(2), \cdots, y^{\mathrm{T}}(j)) \tag{6.75}$$

那么，根据

$$\hat{x}(k \mid j) = E[x(k) \mid y_i] \tag{6.76}$$

有 $x(k)$ 的最小均方预测值

$$\begin{aligned}\hat{x}(k \mid j) &= E[x(k) \mid y_i] \\ &= E[x(k) \mid y(1), y(2), \cdots, y(j)] \\ &= \phi(k,j)E[x(j) \mid y_j] + \sum_{i=j+1}^{k} \phi(k,i)\eta(i-1)E[\omega(i-1) \mid y_i]\end{aligned} \tag{6.77}$$

当 $k \geqslant j$ 时,根据

$$\begin{aligned}&E[x(k) - \bar{x}(k)][\omega(k) - \bar{\omega}(k)]^{\mathrm{T}} = 0 \\ &E[\omega(k) - \bar{\omega}(k)][x(k) - \bar{x}(k)]^{\mathrm{T}} = 0 \\ &E[x(j) - \bar{x}(j)][\omega(k) - \bar{\omega}(k)]^{\mathrm{T}} = 0 \\ &E[\omega(k) - \bar{\omega}(k)][x(j) - \bar{x}(j)]^{\mathrm{T}} = 0\end{aligned} \tag{6.78}$$

有

$$E[\omega(k) \mid \boldsymbol{y}(1), \boldsymbol{y}(2), \cdots, \boldsymbol{y}(j)] = E[\omega(k)] = 0 \tag{6.79}$$

故有状态预测公式

$$\hat{x}(k \mid j) = \phi(k,j)\, \hat{x}(j \mid j) \tag{6.80}$$

考虑到最小方差估计乃是一种无偏估计,故上述的预测误差的方差可写作

$$\begin{aligned}P(k \mid j) &= E[x(k) - \hat{x}(k \mid j)][x(k) - \hat{x}(k \mid j)]^{\mathrm{T}} \\ &= E\Big\{\phi(k,j)[x(j) - \hat{x}(j \quad j)] + \sum_{i=j+1}^{k} \phi(k,j)\eta(i-1)\omega(i-1)\Big\} \\ &\quad \times \Big\{\phi(k,j)[x(j) - \hat{x}(j \mid j)] + \sum_{i=j+1}^{k} \phi(k,j)\eta(i-1)\omega(i-1)\Big\}^{\mathrm{T}}\end{aligned} \tag{6.81}$$

考虑到 $\{\omega(k), k \in T\}$ 的独立性质,及

$$E[x(j), \omega(i)] = 0 \tag{6.82}$$

在 $j \leqslant i$ 时成立,故有

$$\begin{aligned}P(k \mid j) &= \phi(k \mid j)P(j \mid j)\phi^{\mathrm{T}}(k \mid j) \\ &\quad + \sum_{i=j+1}^{k} \phi(k,i)\eta(i-1)R_1(i-1)\eta^{-\mathrm{T}}(i-1)\phi^{\mathrm{T}}(k,i)\end{aligned} \tag{6.83}$$

式(6.81)与(6.83)表明,欲预测状态及确定状态误差的方差,应解决状态滤波及滤波误差的方差问题。Kalman 滤波公式实际上就是一套递推线性滤波公式。它的出发点是给定 $\hat{x}(0|0)$ 与 $P(0|0)$,每实测出一个 $y(k)$,该递推公式即能利用前一瞬时的滤波结果 $\hat{x}(k-1|k-1)$ 与 $P(k-1|k-1)$ 计算出 k 瞬时的 $\hat{x}(k|k)$ 与 $P(k|k)$。一步预测 $\hat{x}(k|k-1)$ 及误差的方差 $P(k|k-1)$ 是 Kalman 滤波公式的中间结果。

2. Kalman 滤波公式

由式(6.80)与(6.83)知,状态的一步预测为

$$\hat{x}(k \mid k-1) = \phi(k \mid k-1)\,\hat{x}(k-1 \mid k-1) \tag{6.84}$$

一步预测误差的方差为

$$\begin{aligned} P(k \mid k-1) &= \phi(k \mid k-1)P(k-1 \mid k-1)\phi^{\mathrm{T}}(k \mid k-1) \\ &\quad + \eta(k-1)R_1(k-1)\eta^{\mathrm{T}}(k-1) \end{aligned} \tag{6.85}$$

根据 $\boldsymbol{y}_{k-1}^{\mathrm{T}} = (y^{\mathrm{T}}(1), y^{\mathrm{T}}(2), \cdots, y^{\mathrm{T}}(k-1))$ 对 $y(k)$ 所做的最小方差预测的误差称之为新息 $\tilde{y}(k \mid k-1) = \tilde{y}(k)$,根据式(6.48)有

$$\begin{aligned} \tilde{y}(k) &= y(k) - E[y(k) \mid y_{k-1}] \\ &= y(k) - E[\vartheta(k)x(k) \mid y_{k-1}] - E[v(k) \mid y(1), y(2), \cdots, y(k-1)] \\ &= y(k) - \vartheta(k)\,\hat{x}(k \mid k-1) \end{aligned} \tag{6.86}$$

根据最小方差的误差理论,$\tilde{y}(k)$ 与 y_{k-1} 互相独立,因而 $\{\tilde{y}(k), k \in T\}$ 是零均值独立序列。

根据式(6.71),有

$$\begin{aligned} \hat{x}(k \mid k) &= E[x(k) \mid y_k] \\ &= E[x(k) \mid y_{k-1}, y(k)] \\ &= E[x(k) \mid y_{k-1}, \tilde{y}(k)] \\ &= E[x(k) \mid y_{k-1}] + E[x(k) \mid \tilde{y}(k)] - \bar{x}(k) \\ &= \hat{x}(k \mid k-1) + \mathrm{cov}[x(k), \tilde{y}(k)]\mathrm{var}^{-1}[\tilde{y}(k)] \cdot [\tilde{y}(k) - E(\tilde{y}(k))] \\ &= \hat{x}(k \mid k-1) + K(k)[y(k) - \vartheta(k)\,\hat{x}(k \mid k-1)] \end{aligned} \tag{6.87}$$

式(6.87)中的滤波增益阵

$$K(k) = \mathrm{cov}[x(k), \tilde{y}(k)]\mathrm{var}^{-1}[\tilde{y}(k)] \tag{6.88}$$

从式(6.87)可知利用新息 $\tilde{y}(k)$ 对预测值 $\hat{x}(k \mid k-1)$ 进行修正,即可得出滤波值,而 $K(k)$ 为修正系数阵。下面就来求 $K(k)$ 的实用计算式。

$$\begin{aligned} &\mathrm{cov}[x(k), \tilde{y}(k)] \\ &= E[x(k) - \bar{x}(k)][\tilde{y}(k) - E(\tilde{y}(k))]^{\mathrm{T}} \\ &= E[x(k) - \hat{x}(k \mid k-1) + \hat{x}(k \mid k-1) - \bar{x}(k)][\tilde{y}(k)]^{\mathrm{T}} \\ &= E[\tilde{x}(k \mid k-1) + \hat{x}(k \mid k-1)][\tilde{y}(k)]^{\mathrm{T}} - \bar{x}(k)E[\tilde{y}(k)]^{\mathrm{T}} \\ &= E[\tilde{x}(k \mid k-1) + \hat{x}(k \mid k-1)][\vartheta(k)x(k) + v(k) - \vartheta(k)\,\hat{x}(k \mid k-1)]^{\mathrm{T}} \\ &= E[\tilde{x}(k \mid k-1) + \hat{x}(k \mid k-1)][\vartheta(k)\,\tilde{x}(k \mid k-1) + v(k)]^{\mathrm{T}} \end{aligned} \tag{6.89}$$

考虑到

$$\hat{x}(k \mid k-1) = E(x(k) \mid y_{k-1}) \tag{6.90}$$

即$\hat{x}(k|k-1)$是y_{k-1}的线性函数，此外$v(k)$与$y_{k-1}^{\mathrm{T}} = (y^{\mathrm{T}}(1), y^{\mathrm{T}}(2), \cdots, y^{\mathrm{T}}(k-1))$中的所有元素互相独立，因而$v(k)$也必独立于$y_{k-1}$的线性函数，故有

$$E[\hat{x}(k \mid k-1)][v(k)]^{\mathrm{T}} = 0 \tag{6.91}$$

又，$v(k)$与$x(k)$也是互相独立的，而

$$\tilde{x}(k \mid k-1) = x(k) - E[x(k) \mid y_{k-1}] \tag{6.92}$$

故有

$$E[\hat{x}(k \mid k-1)][v(k)]^{\mathrm{T}} = 0 \tag{6.93}$$

考虑到式(6.90)与(6.92)，还可以知道，$\tilde{x}(k|k-1)$与y_{k-1}互相独立，因而它也与y_{k-1}的线性函数互相独立。由于$\hat{x}(k|k-1)$就是y_{k-1}的某个特定线性函数，故有

$$E[\hat{x}(k \mid k-1)][\tilde{x}(k \mid k-1)] = 0 \tag{6.94}$$

记

$$P(k \mid k-1) = E[\tilde{x}(k \mid k-1)][\tilde{x}(k \mid k-1)]^{\mathrm{T}} = 0 \tag{6.95}$$

则有

$$\mathrm{cov}[x(k), \tilde{y}(k)] = P(k \mid k-1)\vartheta^{\mathrm{T}}(k) \tag{6.96}$$

又

$$\begin{aligned}
&\mathrm{var}[\tilde{y}(k)] \\
&= E[y(k) - \vartheta(k)\hat{x}(k \mid k-1)][y(k) - \vartheta(k)\hat{x}(k \mid k-1)]^{\mathrm{T}} \\
&= E[\vartheta(k)\tilde{x}(k \mid k-1) + v(k)][\vartheta(k)\tilde{x}(k \mid k-1) + v(k)]^{\mathrm{T}} \\
&= \vartheta(k)P(k \mid k-1)\vartheta^{\mathrm{T}}(k) + R_2(k)
\end{aligned} \tag{6.97}$$

将式(6.96)、(6.97)代入式(6.88)，得

$$K(k) = P(k \mid k-1)\vartheta^{\mathrm{T}}(k)[\vartheta(k)P(k \mid k-1)\vartheta^{\mathrm{T}}(k) + R_2(k)]^{-1} \tag{6.98}$$

现在确定滤波误差的方差

$$P(k \mid k) = E[\tilde{x}(k \mid k)][\tilde{x}(k \mid k)]^{\mathrm{T}} \tag{6.99}$$

考虑到

$$\begin{aligned}
&\tilde{x}(k \mid k) \\
&= x(k) - \hat{x}(k \mid k) \\
&= \tilde{x}(k \mid k-1) - K(k)[y(k) - \vartheta(k)\hat{x}(k \mid k-1)] \\
&= [I - K(k)\vartheta(k)]\tilde{x}(k \mid k-1) - K(k)v(k)
\end{aligned} \tag{6.100}$$

有

$$
\begin{aligned}
&P(k \mid k) \\
&=[I-K(k)\vartheta(k)]P(k \mid k-1)[I-K(k)\vartheta(k)]^{\mathrm{T}}+K(k)R_2(k)K^{\mathrm{T}}(k) \\
&=[I-K(k)\vartheta(k)]P(k \mid k-1)-P(k \mid k-1)\vartheta^{\mathrm{T}}(k)K^{\mathrm{T}}(k) \\
&\quad +K(k)\vartheta(k)P(k \mid k-1)\vartheta^{\mathrm{T}}(k)K^{\mathrm{T}}(k)+K(k)R_2(k)K^{\mathrm{T}}(k) \\
&=[I-K(k)\vartheta(k)]P(k \mid k-1)-P(k \mid k-1)\vartheta^{\mathrm{T}}(k)K^{\mathrm{T}}(k) \\
&\quad +K(k)[\vartheta(k)P(k \mid k-1)\vartheta^{\mathrm{T}}(k)+R_2(k)]K^{\mathrm{T}}(k)
\end{aligned} \tag{6.101}
$$

由式(6.98)知

$$
K(k)[\vartheta(k)P(k \mid k-1)\vartheta^{\mathrm{T}}(k)+R_2(k)]=P(k \mid k-1)\vartheta^{\mathrm{T}}(k) \tag{6.102}
$$

将此式代入上式得

$$
P(k \mid k)=[I-K(k)\vartheta(k)]P(k \mid k-1) \tag{6.103}
$$

式(6.84)、(6.85)、(6.87)、(6.98)、(6.103)给出了一套递推求算$\hat{x}(k \mid k-1)$、$P(k|k-1)$、$K(k)$、$\hat{x}(k|k)$、$P(k|k)$的公式。所需要的数据$\phi(k,k-1)$、$\eta(k)$、$\vartheta(k)$、$R_1(k)$、$R_2(k)$应由给定的系统状态模型与测量模型提供。而实测数据$y_k^{\mathrm{T}}=[y^{\mathrm{T}}(1),y^{\mathrm{T}}(2),\cdots,y^{\mathrm{T}}(k)]$则应实时地由测量仪器提供。作为递推公式,尚缺滤波的初值$\hat{x}(0|0)$与$P(0|0)$,现在讨论初值的选定问题。

为解决初值问题,先考虑一下滤波误差的均值

$$
\begin{aligned}
E[\tilde{x}(k \mid k)] &= E[x(k)-\hat{x}(k \mid k)] \\
&=\phi(k,k-1)E[x(k-1)]-E\{\phi(k,k-1)\hat{x}(k-1 \mid k-1) \\
&\quad +K(k)[\vartheta(k)x(k)+v(k)-\vartheta(k)\hat{x}(k \mid k-1)]\} \\
&=\phi(k,k-1)E[\tilde{x}(k-1 \mid k-1)] \\
&\quad -K(k)\vartheta(k)\phi(k,k-1)E[x(k-1)-\hat{x}(k-1 \mid k-1)] \\
&=[I-K(k)\vartheta(k)]\phi(k,k-1)E[\tilde{x}(k-1 \mid k-1)]
\end{aligned} \tag{6.104}
$$

此式表明,为保证k瞬时滤波无偏,即$E[\tilde{x}(k|k)]=0$,就必须保证$k-1$瞬时滤波无偏,即$E[\tilde{x}(k-1|k-1)]=0$。依次类推,很显然,只要$k=0$瞬时滤波值无偏,就可保证所有瞬时的滤波无偏。欲要求初值无偏,即

$$
E[x(0)-\hat{x}(0 \mid 0)]=0 \tag{6.105}
$$

可取

$$
\hat{x}(0 \mid 0)=E[x(0)]=m \tag{6.106}
$$

此时

$$P(0\mid 0) = E[x(0) - m][x(0) - m]^{\mathrm{T}} = P_0 \tag{6.107}$$

而 m 与 P_0 均为系统状态方程的已知条件。

Kalman 滤波公式集中表示如下

$$\begin{cases} \hat{x}(k \mid k) = \hat{x}(k \mid k-1) + K(k)[y(k) - \vartheta(k)\hat{x}(k \mid k-1)] \\ \hat{x}(k \mid k-1) = \phi(k \mid k-1)\hat{x}(k-1 \mid k-1) \\ K(k) = P(k \mid k-1)\vartheta^{\mathrm{T}}(k)[\vartheta(k)P(k \mid k-1)\vartheta^{\mathrm{T}}(k) + R_2(k)]^{-1} \\ P(k \mid k) = [I - K(k)\vartheta(k)]P(k \mid k-1) \\ P(k \mid k-1) = \phi(k \mid k-1)P(k-1 \mid k-1)\phi^{\mathrm{T}}(k,k-1) \\ \qquad\qquad + \eta(k-1)R_1(k-1)\eta^{\mathrm{T}}(k-1) \end{cases} \tag{6.108}$$

为了使用方便，现将 Kalman 滤波公式组成如图 6.5 所示的计算框图。图中 N 为滤波总步数。按照图 6.5 所示的计算框图编好计算程序后，只要一个循环的计算时间小于离散输入的时间间隔，Kalman 滤波即可不断地、在线地计算下去。易于发现 $P(k\mid k-1)$、$K(k)$、$P(k\mid k)$并不依赖实测值 $y(k)$，只要$\hat{x}(0\mid 0)$与 $P(0\mid 0)$一定，它们都可在实测值出现前计算好，而事先将它们记入计算机，因此称

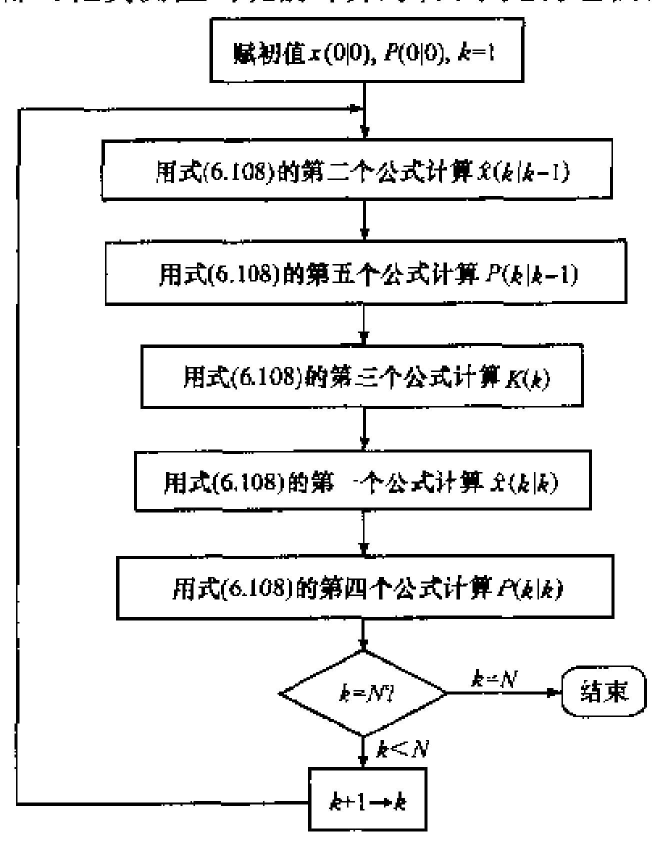


图 6.5　Kalman 滤波程序框图

为离线计算。离线计算虽可能多占用计算机的存储单元，但它可明显地减少在线计算的时间，而这对实时的系统而言，是很有价值的。

引用自动控制理论中结构框图的概念，可以很方便地给出 Kalman 滤波器的结构框图，如图 6.6 所示。它形象地表示了系统、观察器与滤波器间运动的联系。

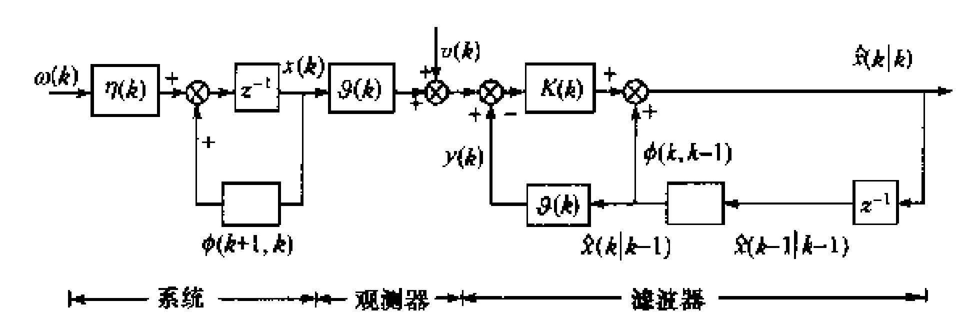


图 6.6 Kalman 滤波器的结构框图

例 6.2 对某定点测距 $k+1$ 次，得实测值为 $y(0), y(1), \cdots, y(k)$。若每次测量都是独立进行的，且距离误差服从正态分布 $\mathcal{N}(0,\sigma)$，试用 Kalman 滤波理论估计每次实测后的距离，并给出相应的估计误差的方差。

解 设到该定点的距离为 x，则

$$\begin{cases} x(k+1) = x(k) \\ y(k) = x(k) + v(k) \end{cases}$$

且

$$\mathrm{cov}[v(k), v(j)] = \sigma^2 \delta_{kj}$$

考虑到选取

$$\hat{x}(0 \mid 0) = y(0)$$

亦有

$$E[x(0) - \hat{x}(0 \mid 0)] = E[x - x - v(0)] = 0$$

仍能保证无偏估计。此时

$$P(0 \mid 0) = \mathrm{var}[x - y(0)] = \sigma^2$$

考虑到 $\phi(k, k-1) = 1, R_1(k) = 0, \vartheta(k) = 1$，可得

$$P(k \mid k-1) = P(k-1 \mid k-1)$$

$$K(k) = P(k \mid k-1)[P(k \mid k-1) + \sigma^2]^{-1}$$

$$P(k \mid k) = [1 - K(k)]P(k \mid k-1) = \sigma^2 K(k)$$

显然

$$P(1 \mid 0) = \sigma^2, \qquad K(1) = \frac{1}{2}$$

$$P(1 \mid 1) = P(2 \mid 1) = \frac{1}{2}\sigma^2, \qquad K(2) = \frac{1}{3}$$

$$P(2 \mid 2) = P(3 \mid 2) = \frac{1}{3}\sigma^2, \qquad K(3) = \frac{1}{4}$$

$$P(3 \mid 3) = P(4 \mid 3) = \frac{1}{4}\sigma^2$$

且有

$$\begin{aligned}
\hat{x}(1 \mid 1) &= \hat{x}(0 \mid 0) + K(1)[y(1) - \hat{x}(0 \mid 0)] \\
&= \frac{1}{2}[y(0) + y(1)]
\end{aligned}$$

$$\begin{aligned}
\hat{x}(2 \mid 2) &= \hat{x}(1 \mid 1) + K(2)[y(2) - \hat{x}(1 \mid 1)] \\
&= \frac{1}{3}[y(0) + y(1) + y(2)]
\end{aligned}$$

$$\begin{aligned}
\hat{x}(3 \mid 3) &= \hat{x}(2 \mid 2) + K(3)[y(3) - \hat{x}(2 \mid 2)] \\
&= \frac{1}{4}[y(0) + y(1) + y(2) + y(3)]
\end{aligned}$$

其实,就本例而言

$$\begin{aligned}
\frac{1}{P(k \mid k)} &= \frac{1}{P(k-1 \mid k-1)} + \frac{1}{\sigma^2} \\
&= \frac{1}{P(k-2 \mid k-2)} + \frac{1}{\sigma^2} + \frac{1}{\sigma^2} = \frac{1}{P(0 \mid 0)} + \frac{k}{\sigma^2}
\end{aligned}$$

故有

$$P(k \mid k) = \frac{1}{k+1}\sigma^2, \qquad K(k) = \frac{1}{k+1}$$

而

$$\begin{aligned}
\hat{x}(k \mid k) &= \hat{x}(k-1 \mid k-1) + K(k)[y(k) - \hat{x}(k-1 \mid k-1)] \\
&= [1 - K(k)]\,\hat{x}(k-1 \mid k-1) + K(k)y(k) \\
&= \frac{k}{k+1}\,\hat{x}(k-1 \mid k-1) + \frac{1}{k+1}y(k) \\
&= \frac{k}{k+1}\left[\frac{k-1}{k}\hat{x}(k-2 \mid k-2) + \frac{1}{k}y(k-1)\right] + \frac{1}{k+1}y(k) \\
&= \frac{k-1}{k+1}\left[\frac{k-2}{k-1}\,\hat{x}(k-3 \mid k-3) + \frac{1}{k-1}y(k-2)\right] \\
&\quad + \frac{1}{k+1}[y(k) + y(k-1)]
\end{aligned}$$

$$= \frac{1}{k+1}[y(k) + y(k-1) + \cdots + y(0)]$$

仅就本例而言，不论 $P(0|0)$ 取值多少，包括 $P(0|0)=\infty$，只要 $k\to\infty$，就有 $P(k|k)\to 0$。因而

$$\lim_{k\to\infty} \hat{x}(k|k) = x$$

也就是说，只要 k 足够大，$\hat{x}(0|0)$ 取任何值都是无关紧要的。

6.4　模型参考自适应辨识方法

“模型”这个概念已广泛用于自适应控制中。如果把控制系统希望达到的控制性能指标用一个称作参考模型的理想化控制系统的性能来描述，并以 $\tilde{z}_m(k)$ 表示每一瞬间实际过程与参考模型之间的特性差异，那么根据这个差异 $\tilde{z}_m(k)$，不断地修改控制器的参数，就可使实际系统的性能指标尽可能地接近参考模型。这种控制思想就叫做模型参考自适应控制，其基本结构如图 6.7 所示。

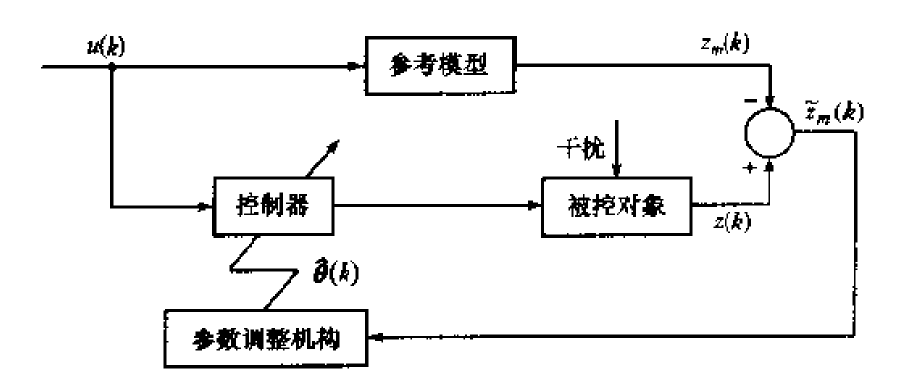


图 6.7　模型参考自适应原理

同样，“模型参考”的概念也可以用于实时在线辨识[81]。这时，与模型参考自适应控制相反，被辨识的对象扮演了参考模型的角色，如图 6.8 所示。当可调模型结构已知时，根据可调模型与参考模型(实际过程)之间的输出偏差 $\tilde{z}_m(k)$ 进行适当的运算，通常用积分或积分与比例运算，并根据运算结果不断地修改可调模型的参数，使在一组已知的输入作用下，可调模型的输出尽可能地接近参考模型的输出。当可调模型与参考模型之间的差别无法进行改善时，可调模型的参数就是实际过程参数的估计值。

模型参考自适应辨识方法，就其基本结构来说，可分成并联、串并联和串联三种类型。这里只讨论并联型的模型参考法，亦称输出误差模型参考法。

设实际系统可用如下模型描述

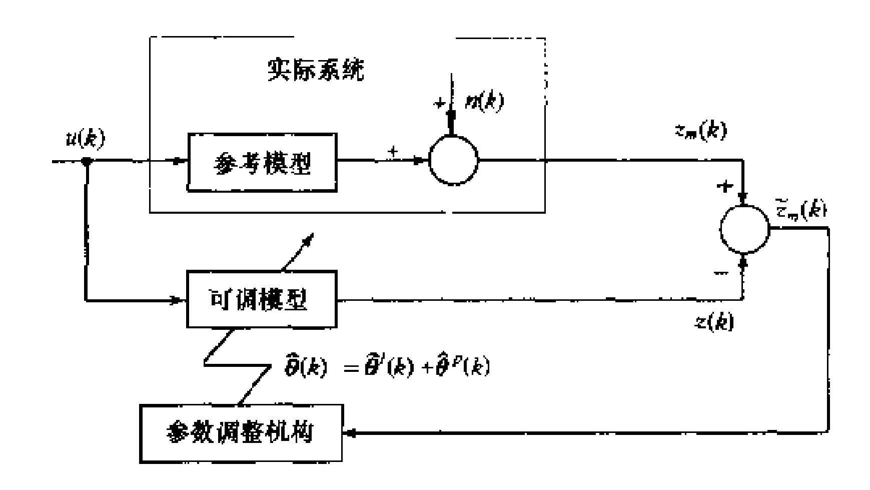


图 6.8 模型参考自适应辨识原理

$$z(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k) + n(k) \tag{6.109}$$

式中，$u(k)$和 $z(k)$是系统的输入和输出变量；$n(k)$是系统的附加噪声；且

$$\begin{cases} A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a} \\ B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b} \end{cases} \tag{6.110}$$

设阶次 n_a 和 n_b 已知，并置

$$\begin{cases} \boldsymbol{\theta}_0 = [\boldsymbol{a}_0^{\mathrm{T}}, \boldsymbol{b}_0^{\mathrm{T}}]^{\mathrm{T}} = [a_1, \cdots, a_{n_a}, b_1, \cdots, b_{n_b}]^{\mathrm{T}} \\ \boldsymbol{h}(k) = [-\boldsymbol{z}^{\mathrm{T}}(k), \boldsymbol{u}^{\mathrm{T}}(k)]^{\mathrm{T}} = [-z(k-1), \cdots, \\ \qquad -z(k-n_a), u(k-1), \cdots, u(k-n_b)]^{\mathrm{T}} \end{cases} \tag{6.111}$$

则参考模型可写成最小二乘格式

$$z(k) = \boldsymbol{\theta}_0^{\mathrm{T}} \boldsymbol{h}(k) + e(k) \tag{6.112}$$

式中，$e(k)$为有色噪声，可表示成

$$e(k) = A(z^{-1}) n(k) \tag{6.113}$$

为了方便起见，先考虑 $n(k)=0$ 的情况。根据理论分析，其结果也适用于 $n(k) \neq 0$的情况。

考虑采用如下的可调模型

$$z_m(k) = \frac{\hat{B}(z^{-1})}{\hat{A}(z^{-1})} u(k) \tag{6.114}$$

式中，$u(k)$和 $z_m(k)$是可调模型输入和输出变量。置

$$\begin{cases}\hat{\boldsymbol{\theta}}(k) = [\hat{\boldsymbol{a}}^{\mathrm{T}}(k), \hat{\boldsymbol{b}}^{\mathrm{T}}(k)]^{\mathrm{T}} = [\hat{a}_1(k), \cdots, \hat{a}_{n_a}(k), \hat{b}_1(k), \cdots, \hat{b}_{n_b}(k)]^{\mathrm{T}} \\ \boldsymbol{h}_m(k) = [-\boldsymbol{z}_m{}^{\mathrm{T}}(k), \boldsymbol{u}^{\mathrm{T}}(k)]^{\mathrm{T}} = [-z_m(k-1), \cdots, \\ \qquad - z_m(k-n_a), u(k-1), \cdots, u(k-n_b)]^{\mathrm{T}}\end{cases} \tag{6.115}$$

并定义可调模型的先验和后验广义误差为

$$\begin{cases}\text{先验}: \tilde{z}_m^0(k) = z(k) - z_m^0(k) \\ \text{后验}: \tilde{z}_m(k) = z(k) - z_m(k)\end{cases} \tag{6.116}$$

式中

$$\begin{cases}z(k) = \boldsymbol{\theta}_0^{\mathrm{T}} \boldsymbol{h}(k) \\ z_m(k) = \hat{\boldsymbol{\theta}}^{\mathrm{T}}(k) \boldsymbol{h}_m(k) = [\hat{\boldsymbol{\theta}}^l(k) + \hat{\boldsymbol{\theta}}^p(k)]^{\mathrm{T}} \boldsymbol{h}_m(k) \\ z_m^0(k) = [\hat{\boldsymbol{\theta}}(k-1)]^{\mathrm{T}} \boldsymbol{h}_m(k)\end{cases} \tag{6.117}$$

上式中，$\hat{\boldsymbol{\theta}}^l(k)$和 $\hat{\boldsymbol{\theta}}^p(k)$分别对应图 6.8 所示的参数调整机构的积分和比例运算输出。也就是说，图 6.8 中的参数调整机构选用如下运算规则

$$\begin{cases}\text{积分作用}: \hat{\boldsymbol{\theta}}^l(k) = \sum_{l=0}^{k} \boldsymbol{\varphi}_1(\varepsilon(l)) + \hat{\boldsymbol{\theta}}^l(-1) = \hat{\boldsymbol{\theta}}^l(k-1) + \boldsymbol{\varphi}_1(\varepsilon(k)) \\ \text{比例作用}: \hat{\boldsymbol{\theta}}^p(k) = \boldsymbol{\varphi}_2(\varepsilon(k)) \\ \text{综合}: \hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}^l(k) + \hat{\boldsymbol{\theta}}^p(k)\end{cases} \tag{6.118}$$

式中，$\boldsymbol{\varphi}_1$ 和 $\boldsymbol{\varphi}_2$ 是 $\varepsilon(\cdot)$的待定函数，$\boldsymbol{\varphi}_1$ 和 $\boldsymbol{\varphi}_2$ 的结构将决定辨识算法的具体形式；且 $\varepsilon(k)$必须满足下列关系

$$\varepsilon(k) = \tilde{z}_m(k) + \sum_{i=1}^{n_a} \beta_i \tilde{z}_m(k-i) \tag{6.119}$$

如果系数 β_i 使下列脉冲传递函数

$$H(z) = \frac{1 + \sum_{i=1}^{n_a} \beta_i z^{-i}}{1 + \sum_{i=1}^{n_a} a_i z^{-i}} - \frac{1}{2} \tag{6.120}$$

是严格正实的，即脉冲传递函数 $H(z)$的所有极点都位于 z 平面的单位圆内，且

$$\mathrm{Re}[H(\mathrm{e}^{\mathrm{j}\omega})] > 0, \forall \omega$$

及当 z 为实数时，$H(z)$也是实数，那么辨识结果$\hat{\boldsymbol{\theta}}(k)$将保证$\lim_{k\to\infty} \tilde{z}_m(k) = 0$。系数 β_i 为待定参数，它依赖于系统模型的真实参数 a_i。

6.4.1 φ_1 和 φ_2 的确定

把式(6.117)代入式(6.116),并利用式(6.111)和(6.115),可得

$$\tilde{z}_m(k) = -\boldsymbol{a}_0^{\mathrm{T}}\tilde{\boldsymbol{z}}_m(k) + \eta(k) \tag{6.121}$$

式中

$$\begin{cases}\tilde{\boldsymbol{z}}_m(k) = [\tilde{z}_m(k-1),\cdots,\tilde{z}_m(k-n_a)]^{\mathrm{T}} \\ \eta(k) = -[\hat{\boldsymbol{\theta}}(k) - \boldsymbol{\theta}_0]^{\mathrm{T}}\boldsymbol{h}_m(k)\end{cases} \tag{6.122}$$

令

$$\eta^*(k) = -\eta(k) = [\hat{\boldsymbol{\theta}}(k) - \boldsymbol{\theta}_0]^{\mathrm{T}}\boldsymbol{h}_m(k) \tag{6.123}$$

并注意到式(6.118),则

$$\eta^*(k) = \left\{\sum_{l=0}^{k}[\boldsymbol{\varphi}_1^{\mathrm{T}}(\varepsilon(l)) + \boldsymbol{\varphi}_2^{\mathrm{T}}(\varepsilon(l))] + \bar{\boldsymbol{\theta}}^{\mathrm{T}}\right\}\boldsymbol{h}_m(k) \tag{6.124}$$

式中

$$\bar{\boldsymbol{\theta}} = -\boldsymbol{\theta}_0 + \hat{\boldsymbol{\theta}}^l(-1) \tag{6.125}$$

同时将式(6.119)写成

$$\varepsilon(k) = \tilde{z}_m(k) + \boldsymbol{\beta}^{\mathrm{T}}\tilde{\boldsymbol{z}}_m(k) \tag{6.126}$$

式中

$$\boldsymbol{\beta} = [\beta_1,\beta_2,\cdots,\beta_{n_a}]^{\mathrm{T}} \tag{6.127}$$

于是,并联型的模型参考自适应辨识问题转化成设计 $\boldsymbol{\varphi}_1$ 和 $\boldsymbol{\varphi}_2$,使下列以 $\eta(k)$ 和 $\varepsilon(k)$ 为输入输出变量的反馈系统

$$\begin{cases}\tilde{z}_m(k) = -\boldsymbol{a}_0^{\mathrm{T}}\tilde{\boldsymbol{z}}_m(k) + \eta(k) \\ \varepsilon(k) = \tilde{z}_m(k) + \boldsymbol{\beta}^{\mathrm{T}}\tilde{\boldsymbol{z}}_m(k) \\ \eta^*(k) = -\eta(k) = \left\{\sum_{l=0}^{k}[\boldsymbol{\varphi}_1^{\mathrm{T}}(\varepsilon(l)) + \boldsymbol{\varphi}_2^{\mathrm{T}}(\varepsilon(l))] + \bar{\boldsymbol{\theta}}^{\mathrm{T}}\right\}\boldsymbol{h}_m(k)\end{cases} \tag{6.128}$$

是渐进稳定的。根据Popov超稳定性定理,如果 $\boldsymbol{\beta}$ 使得脉冲传递函数 $H(z)$ 是严格正实的,且 $\boldsymbol{\varphi}_1$ 和 $\boldsymbol{\varphi}_2$ 使Popov不等式成立,即

$$\sum_{k=0}^{k_1}\varepsilon(k)\eta^*(k) \geqslant -r_0^2,\ \forall k_1 \geqslant 0 \tag{6.129}$$

也就是

$$\sum_{k=0}^{k_1}\varepsilon(k)\left\{\sum_{l=0}^{k}[\boldsymbol{\varphi}_1^{\mathrm{T}}(\varepsilon(l)) + \boldsymbol{\varphi}_2^{\mathrm{T}}(\varepsilon(l))] + \bar{\boldsymbol{\theta}}^{\mathrm{T}}\right\}\boldsymbol{h}_m(k) \geqslant -r_0^2,\ \forall k_1 \geqslant 0 \tag{6.130}$$

式中,r_0^2 是任意有限正常数,那么式(6.128)反馈系统是渐进稳定的。同时,不等

式(6.130)的解为

$$\begin{cases}\boldsymbol{\varphi}_1^{\mathrm{T}}(\varepsilon(l)) = [\boldsymbol{P}(k-l)\boldsymbol{h}_m(l)]^{\mathrm{T}}\varepsilon(l)\\ \boldsymbol{\varphi}_2(\varepsilon(k)) = [\boldsymbol{Q}\boldsymbol{h}_m(k)]^{\mathrm{T}}\varepsilon(k)\end{cases} \tag{6.131}$$

式中,$\boldsymbol{P}(\cdot)$和 $\boldsymbol{Q}$ 均为正定对称阵。若定义 $\boldsymbol{P}(0)=\boldsymbol{P}$(常数阵),则上式写成

$$\begin{cases}\boldsymbol{\varphi}_1(\varepsilon(l)) = \boldsymbol{P}\boldsymbol{h}_m(l)\varepsilon(l)\\ \boldsymbol{\varphi}_2(\varepsilon(k)) = \boldsymbol{Q}\boldsymbol{h}_m(k)\varepsilon(k)\end{cases} \tag{6.132}$$

上述分析表明,式(6.132)使得反馈系统(6.128)是渐进稳定的,将式(6.132)代入式(6.118),则可调节模型的参数可按下列式子调整

$$\begin{cases}\text{积分作用}:\hat{\boldsymbol{\theta}}^{l}(k) = \hat{\boldsymbol{\theta}}^{l}(k-1) + \boldsymbol{P}\boldsymbol{h}_m(k)\varepsilon(k)\\ \text{比例作用}:\hat{\boldsymbol{\theta}}^{p}(k) = \boldsymbol{Q}\boldsymbol{h}_m(k)\varepsilon(k)\\ \text{综合}:\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}^{l}(k) + \hat{\boldsymbol{\theta}}^{p}(k)\end{cases} \tag{6.133}$$

但是,在 k 时刻,$\varepsilon(k)$仍无法计算,为此还需要研究 $\varepsilon(k)$的计算问题。

6.4.2 $\boldsymbol{\varepsilon}(k)$的计算

根据式(6.119)的定义,因 $\varepsilon(k)$与后验广义误差$\tilde{z}_m(k)$有关,而计算$\tilde{z}_m(k)$需要用到$\hat{\boldsymbol{\theta}}(k)$,故在 k 时刻$\varepsilon(k)$是无法计算的。

仿照式(6.119),定义一个新变量

$$\varepsilon^0(k) = \tilde{z}_m^0(k) + \sum_{i=1}^{n_a}\beta_i\tilde{z}_m(k-i) \tag{6.134}$$

因先验广义误差$\tilde{z}_m^0(k)$的计算只需要$\hat{\boldsymbol{\theta}}^{l}(k-1)$,故在 k 时刻$\varepsilon^0(k)$是可以计算的。

比较式(6.119)和(6.134),并利用式(6.116)、(6.117)和(6.133),可得

$$\begin{aligned}\varepsilon^0(k)-\varepsilon(k) &= \tilde{z}_m^0(k)-\tilde{z}_m(k) = z_m(k)-z_m^0(k)\\ &=[\hat{\boldsymbol{\theta}}^{l}(k)+\hat{\boldsymbol{\theta}}^{p}(k)-\hat{\boldsymbol{\theta}}^{l}(k-1)]^{\mathrm{T}}\boldsymbol{h}_m(k)\\ &=\boldsymbol{h}_m^{\mathrm{T}}(k)(\boldsymbol{P}+\boldsymbol{Q})\boldsymbol{h}_m(k)\varepsilon(k)\end{aligned} \tag{6.135}$$

则

$$\varepsilon(k) = \frac{\varepsilon^0(k)}{1+\boldsymbol{h}_m^{\mathrm{T}}(k)(\boldsymbol{P}+\boldsymbol{Q})\boldsymbol{h}_m(k)} \tag{6.136}$$

式中

$$\varepsilon^0(k) = z(k) - [\hat{\boldsymbol{\theta}}^{l}(k-1)]^{\mathrm{T}}\boldsymbol{h}_m(k) + \sum_{i=1}^{n_a}\beta_i\tilde{z}_m(k-i) \tag{6.137}$$

式(6.136)即为计算 $\varepsilon(k)$的公式。

6.4.3 A 辨识算法类

综上所述,并联模型参考自适应 A 类算法 RMR-A(recursive model reference adaptive algorithm-A)可归纳为

$$\begin{cases}\hat{\boldsymbol{\theta}}(k)=\hat{\boldsymbol{\theta}}^{I}(k)+\hat{\boldsymbol{\theta}}^{P}(k)\\ \hat{\boldsymbol{\theta}}^{I}(k)=\hat{\boldsymbol{\theta}}^{I}(k-1)+\dfrac{\boldsymbol{P}\boldsymbol{h}_m(k)\varepsilon^0(k)}{1+\boldsymbol{h}_m^{\mathrm{T}}(k)(\boldsymbol{P}+\boldsymbol{Q})\boldsymbol{h}_m(k)}\\ \hat{\boldsymbol{\theta}}^{P}(k)=\dfrac{\boldsymbol{Q}\boldsymbol{h}_m(k)\varepsilon^0(k)}{1+\boldsymbol{h}_m^{\mathrm{T}}(k)(\boldsymbol{P}+\boldsymbol{Q})\boldsymbol{h}_m(k)}\\ \boldsymbol{Q}=\alpha\boldsymbol{P},\alpha\geqslant-0.5\quad 或\quad \boldsymbol{Q}+\dfrac{1}{2}\boldsymbol{P}\geqslant\boldsymbol{0}\\ \varepsilon^0(k)=z(k)-[\hat{\boldsymbol{\theta}}^{I}(k-1)]^{\mathrm{T}}\boldsymbol{h}_m(k)+\sum\limits_{i=1}^{n_a}\beta_i\tilde{z}_m(k-i)\\ \tilde{z}_m(k)=z(k)-\hat{\boldsymbol{\theta}}^{\mathrm{T}}(k)\boldsymbol{h}_m(k)\end{cases}\tag{6.138}$$

式中,$\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 是任意的正定对称常数阵;系数 β_i 使式(6.120)的脉冲传递函数 $H(z)$是严格正实的。

A 类辨识算法的几点说明:

(1) 当 $\boldsymbol{P}\neq\boldsymbol{0},\boldsymbol{Q}=\boldsymbol{0}$ 时,算法为比例积分型;当 $\boldsymbol{P}=\boldsymbol{0},\boldsymbol{Q}\neq\boldsymbol{0}$时,算法为积分型。

(2) 数据向量 $\boldsymbol{h}_m(k)$不含参考模型(实际过程)的输出量 $z(k)$。

(3) 如果 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 阵都是正定的,则有利于输出广义误差的收敛性。如果 $\boldsymbol{P}$ 是正实的,而 $\boldsymbol{Q}$ 是满足(6.138)第 4 式的负定阵,则有利于参数的收敛性。$\boldsymbol{Q}+\frac{1}{2}\boldsymbol{P}\geqslant\boldsymbol{0}$(即 $\boldsymbol{Q}+\frac{1}{2}\boldsymbol{P}$ 是正定阵)是参数收敛的条件。

(4) 原则上说,$\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 的选择是任意的。但是,对实际问题来说,需要视具体情况综合考虑各种因素,最后选定 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 阵。一般来说,增加 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 阵的值将有利于辨识的收敛速度,但不利于辨识的精度。反之,有利于辨识的精度,但不利于辨识的速度。可见,$\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 阵的大小对收敛速度和辨识精度的要求是矛盾的。因此,$\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 的具体选择必须在收敛速度和精度之间做出折中。鉴于上述原因,如果让 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 阵是时变递减的,即当 k 比较小时,$\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 阵取值比较大,以加快收敛速度,随着 k 的增加,$\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 阵值不断递减,以保证辨识精度,这样就解决了收敛速度和精度之间的矛盾。这种考虑就是下面所给的 B 类辨识算法。

(5) A 类辨识算法适用于时变对象,由于 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 阵为常数阵,因此算法任何时候都具有较强的跟踪能力。B 类辨识算法只适用于时不变对象,因为 B 类辨识

算法中的 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 阵随时间的推移将慢慢趋于零。然而，在有噪声的情况下，A 类辨识算法一般只能获得有偏估计，而 B 类辨识算法可获得无偏估计。

6.4.4 B 类辨识算法

A 类辨识算法中 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 改用时变矩阵后的算法就是并联型的模型参考自适应 B 类辨识算法 RMR-B(recursive model reference adaptive algorithm-B)

$$
\begin{cases}
\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}^{I}(k) + \hat{\boldsymbol{\theta}}^{P}(k) \\
\hat{\boldsymbol{\theta}}^{I}(k) = \hat{\boldsymbol{\theta}}^{I}(k-1) + \dfrac{\boldsymbol{P}(k-1)\boldsymbol{h}_m(k)\varepsilon^0(k)}{1+\boldsymbol{h}_m^{\mathrm{T}}(k)[\boldsymbol{P}(k-1)+\boldsymbol{Q}(k-1)]\boldsymbol{h}_m(k)} \\
\hat{\boldsymbol{\theta}}^{P}(k) = \dfrac{\boldsymbol{Q}(k-1)\boldsymbol{h}_m(k)\varepsilon^0(k)}{1+\boldsymbol{h}_m^{\mathrm{T}}(k)[\boldsymbol{P}(k-1)+\boldsymbol{Q}(k-1)]\boldsymbol{h}_m(k)} \\
\boldsymbol{P}(k) = \boldsymbol{P}(k-1) + \dfrac{\boldsymbol{P}(k-1)\boldsymbol{h}_m(k)\boldsymbol{h}_m^{\mathrm{T}}(k)\boldsymbol{P}(k-1)}{1+\boldsymbol{h}_m^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{h}_m(k)}, \boldsymbol{P}(0) > \boldsymbol{0} \\
\boldsymbol{Q} = \alpha\boldsymbol{P}, \alpha \geqslant -0.5 \\
\varepsilon^0(k) = z(k) - [\hat{\boldsymbol{\theta}}^{I}(k-1)]^{\mathrm{T}}\boldsymbol{h}_m(k) + \sum_{i=1}^{n_a}\beta_i \tilde{z}_m(k-i) \\
\tilde{z}_m(k) = z(k) - \hat{\boldsymbol{\theta}}^{\mathrm{T}}(k)\boldsymbol{h}_m(k)
\end{cases}
\tag{6.139}
$$

式中，系数 β_i 使式(6.120)的脉冲传递函数 $H(z)$ 是严格正实的。

B 类辨识算法的几点说明：

(1) 采用 B 类算法，输出后验广义误差 $\tilde{z}_m(k)$ 及参数估计值 $\hat{\boldsymbol{\theta}}(k)$ 是一致收敛的，即

$$
\begin{cases}
\lim\limits_{k\to\infty} \tilde{z}_m(k) = 0 \\
\lim\limits_{k\to\infty} \hat{\boldsymbol{\theta}}(k) = \boldsymbol{0}
\end{cases}
\tag{6.140}
$$

(2) 当 $\alpha \geqslant -0.5$ 时，算法为比例积分型；当 $\alpha=0$ 时，算法为积分型。

(3) α 取负值($-0.5 \leqslant \alpha < 0$)将有利于参数的收敛性；$\alpha$ 取正值则有利于输出广义误差的收敛性。α 取值越大，参数估计值的收敛速度越慢。

6.4.5 C 类辨识算法

不论 A 类还是 B 类辨识算法都需要确定系数 β_i，使式(6.120)的脉冲传递函数 $H(z)$ 是严格正实的。系数 β_i 的选择直接依赖于系统模型的真实参数 a_i。然而，由于参数 a_i 本身是未知的，因此 A 类和 B 类辨识算法在系数 β_i 的选择问题上

将遇到困难。这就引出了 C 类辨识算法。

定义一个增维的可调向量

$$\begin{aligned}\hat{\boldsymbol{\theta}}_E(k) &= [\hat{\boldsymbol{\theta}}^T(k), \hat{\boldsymbol{\beta}}^T(k)]^T \\ &= [\hat{a}_1(k), \cdots, \hat{a}_{n_a}(k), \hat{b}_1(k), \cdots, \hat{b}_{n_b}(k), \hat{\beta}_1(k), \cdots, \hat{\beta}_{n_a}(k)]^T \end{aligned} \tag{6.141}$$

和一个增维的数据向量

$$\begin{aligned}\bar{\boldsymbol{h}}_m(k) &= [\boldsymbol{h}_m^T(k), -\tilde{\boldsymbol{z}}_m^T(k)]^T \\ &= [-z_m(k-1), \cdots, -z_m(k-n_a), u(k-1), \cdots, \\ &\quad u(k-n_b), -\tilde{z}_m(k-1), \cdots, -\tilde{z}_m(k-n_a)]^T \end{aligned} \tag{6.142}$$

这意味着把系数 β_i 也看做辨识参数。同时，式(6.126)和(6.134)分别用下列式子代替

$$\begin{cases} \varepsilon(k) = \tilde{z}_m(k) + \hat{\boldsymbol{\beta}}^T(k)\tilde{\boldsymbol{z}}_m(k) \\ \varepsilon^0(k) = \tilde{z}_m^0(k) + [\hat{\boldsymbol{\beta}}^I(k-1)]^T\tilde{\boldsymbol{z}}_m(k) \end{cases} \tag{6.143}$$

式中

$$\begin{cases} \hat{\boldsymbol{\beta}}(k) = \hat{\boldsymbol{\beta}}^I(k) + \hat{\boldsymbol{\beta}}^P(k) \\ \tilde{z}_m(k) = z(k) - \hat{\boldsymbol{\theta}}_E^T(k)\bar{\boldsymbol{h}}_m(k) \\ \tilde{z}_m^0(k) = z(k) - [\hat{\boldsymbol{\theta}}_E^I(k-1)]^T\bar{\boldsymbol{h}}_m(k) \end{cases} \tag{6.144}$$

式中，$\hat{\boldsymbol{\beta}}^I(k)$和$\hat{\boldsymbol{\beta}}^P(k)$分别对应于积分和比例运算的结果。

经上述增维处理后，C 类辨识算法 RMR-C 可直接从 B 类辨识算法导出

$$\begin{cases} \hat{\boldsymbol{\theta}}_E(k) = \hat{\boldsymbol{\theta}}_E^I(k) + \hat{\boldsymbol{\theta}}_E^P(k) \\ \hat{\boldsymbol{\theta}}_E^I(k) = \hat{\boldsymbol{\theta}}_E^I(k-1) + \dfrac{\bar{\boldsymbol{P}}(k-1)\bar{\boldsymbol{h}}_m(k)\varepsilon^0(k)}{1 + \bar{\boldsymbol{h}}_m^T(k)[\bar{\boldsymbol{P}}(k-1) + \bar{\boldsymbol{Q}}(k-1)]\bar{\boldsymbol{h}}_m(k)} \\ \hat{\boldsymbol{\theta}}_E^P(k) = \dfrac{\bar{\boldsymbol{Q}}(k-1)\bar{\boldsymbol{h}}_m(k)\varepsilon^0(k)}{1 + \bar{\boldsymbol{h}}_m^T(k)[\bar{\boldsymbol{P}}(k-1) + \bar{\boldsymbol{Q}}(k-1)]\bar{\boldsymbol{h}}_m(k)} \\ \bar{\boldsymbol{P}}(k) = \bar{\boldsymbol{P}}(k-1) + \dfrac{\bar{\boldsymbol{P}}(k-1)\bar{\boldsymbol{h}}_m(k)\bar{\boldsymbol{h}}_m^T(k)\bar{\boldsymbol{P}}(k-1)}{1 + \bar{\boldsymbol{h}}_m^T(k)\bar{\boldsymbol{P}}(k-1)\bar{\boldsymbol{h}}_m(k)}, \bar{\boldsymbol{P}}(0) > 0 \\ \bar{\boldsymbol{Q}}(k) = \alpha\bar{\boldsymbol{P}}(k), \alpha \geqslant -0.5 \\ \varepsilon^0(k) = z(k) - [\hat{\boldsymbol{\theta}}_E^I(k-1)]^T\bar{\boldsymbol{h}}_m(k) + [\hat{\boldsymbol{\beta}}^I(k-1)]^T\tilde{\boldsymbol{z}}_m(k) \end{cases} \tag{6.145}$$

当然，C 类辨识算法也可从 A 类辨识算法导出，算法的形式雷同于 A 类算法增维后的结果。

最后需要提醒注意，模型参考自适应辨识算法中所用的数据向量 $\boldsymbol{h}_m(k)$ 或 $\bar{\boldsymbol{h}}_m(k)$ 均不含实际系统的输出 $z(k)$，这一点和以前介绍过的辨识算法是不一样的。另外，虽然上述三类辨识算法都是在无噪声的情况下推导出来的，然而它们也适用于有噪声的情况。对不同的噪声类型，它们的收敛性质是不同的。

6.5 小　　结

Bayes 辨识方法的基本思想是把所要估计的参数看做随机变量，然后设法通过观测与该参数有关联的其他变量，以此来推断这个参数。

极大后验参数估计与极大似然估计有着密切的联系，而出发点又不一样。极大似然估计立足于直接极大化数据的条件概率密度函数，极大后验参数估计则是基于极大化参数的后验概率密度函数。

条件期望参数估计直接以参数的条件数学期望作为参数的估计值，它等价于极小化参数估计误差的方差。不管参数 $\boldsymbol{\theta}$ 的后验概率密度函数取什么形式，条件期望参数估计总是无偏一致估计。但在一般情况下，条件期望参数估计工程上是难以应用的。

Bayes 参数辨识算法与最小二乘递推算法类似。在假设正态分布的前提下，Bayes 估计和 Markov 估计是一致的。它相当于加权最小二乘法中的加权因子取 $\Lambda(k)=1/\sigma_v^2$。

Kalman 滤波器是一种经典的 Bayes 结构。它具有学习功能，或者说，可以根据所得测量数据做出必要的决策，以调整滤波程序，使之适应系统或环境的变化，从而得到接近最优的状态估计。Kalman 滤波公式实际上就是一套递推线性滤波公式。它的出发点是给定的 $\hat{\boldsymbol{x}}(0|0)$ 与 $\boldsymbol{P}(0|0)$，而每实测出一个 $\boldsymbol{y}(k)$，该递推公式即能利用前一瞬时的滤波结果 $\hat{\boldsymbol{x}}(k-1|k-1)$ 与 $\boldsymbol{P}(k-1|k-1)$ 计算出 k 瞬时的 $\hat{\boldsymbol{x}}(k|k)$ 与 $\boldsymbol{P}(k|k)$。一步预测 $\hat{\boldsymbol{x}}(k|k-1)$ 及其误差的方差 $\boldsymbol{P}(k|k-1)$ 是 Kalman 滤波公式的中间结果。

模型参考自适应控制的思想就是，把控制系统希望达到的控制性能指标用一个称作参考模型的理想化控制系统的性能来描述，并根据每一瞬间实际系统与参考模型之间的特性差异，不断地修改控制器的参数，使实际系统的性能指标尽可能地接近参考模型。

习　　题

1. 简述 Bayes 辨识的基本原理。

2. 分析极大后验参数估计方法与条件期望参数估计方法之间的内在联系。

3. 辨别预测、滤波与平滑三个概念，简述 Kalman 滤波的步骤。

4. 什么是模型参考自适应控制？

5. 考虑图 3.6 所示的仿真对象，图中，$v(k)$是服从 $\mathcal{N}(0,1)$分布的不相关随机噪声。且

$$G(z^{-1}) = B(z^{-1})/A(z^{-1}), \quad N(z^{-1}) = D(z^{-1})/C(z^{-1})$$

$$\begin{cases} A(z^{-1}) = 1 - 1.5a_1 z^{-1} + 0.7z^{-2} = C(z^{-1}) \\ B(z^{-1}) = 1.0z^{-1} + 0.5z^{-2} \\ D(z^{-1}) = 1 - z^{-1} + 0.2z^{-2} \end{cases}$$

若仿真对象选择如下的模型结构

$$\begin{aligned} & z(k) + a_1 z(k-1) + a_2 z(k-2) \\ = & b_1 u(k-1) + b_2 u(k-2) + v(k) + d_1 v(k-1) + d_2 v(k-2) \end{aligned}$$

试用 Bayes 辨识算法求出上述模型的参数。

第 7 章　神经网络模型辨识

对于参数易变的非线性复杂系统的模型辨识，前面讨论的古典辨识方法和现代辨识方法显得无能为力，而神经元网络则可以对这类系统进行辨识。因此，本章讨论神经网络模型辨识问题。7.1～7.3 节介绍神经网络辨识的基本概念、神经网络模型辨识中常用结构、以及辨识中常用 BP 网络训练算法等。BP 算法在系统辨识中得到了最广泛的应用，但一般 BP 网络训练时会出现很多问题，诸如不收敛、训练速度慢或精度不高等。因此，7.4 节讨论改进的 BP 网络训练算法，其中包括四种改进算法及其他的网络训练技巧。7.5 节是神经网络辨识的 MATLAB 仿真举例，其中包括对开发的具有噪声二阶系统辨识的 MATLAB 程序剖析、对多维非线性辨识的 MATLAB 程序剖析。7.6 和 7.7 节分别讨论基于改进遗传算法的神经网络和模糊神经网络(FNN)，并列举遗传神经解耦仿真及实验结果，对开发的 FNN 非线性多变量系统的 MATLAB 解耦仿真程序进行剖析。

7.1　神经网络概念与特性

神经元网络是智能控制技术的三大组成部分(神经网络、模糊控制和专家系统)之一。人工神经元是专家根据生物神经元特点在工程上的应用研究发展起来的，它是神经网络的基本处理单元。不同类型的神经网络有各自的激发函数和学习方法。由于神经网络对非线性具有较强的跟踪或自适应能力，因此，神经元网络作为一种新技术引起了人们的巨大兴趣，并越来越多地用于辨识和控制领域。

7.1.1　人工神经元模型

人工神经元是神经网络的基本处理单元，是对生物神经元的简化和模拟。图 7.1 表示了一种简化的神经元结构。从图可见，它是一个多输入、单输出的非线性元件，其输入输出关系可描述为

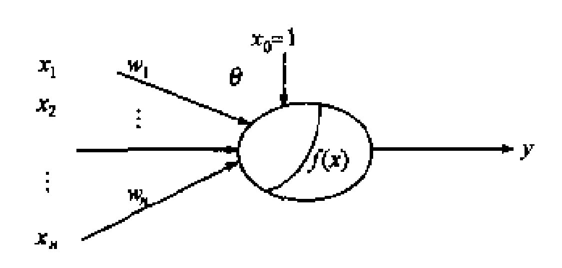


图 7.1　神经元结构模型

$$\begin{cases} I = \sum_{j=1}^{n} w_j x_j - \theta \\ y = f(I) \end{cases} \tag{7.1}$$

式中，$x_j(j=1,2,\cdots,n)$ 是从其他细胞传来的输入信号，θ 为阈值，权系数 w_j 表示连接的强度，说明突触的负载。$f(x)$称为激发函数或作用函数，其非线性特性可用阈值型、分段线性型和连续型激发函数近似。

为了方便，有时将 $-\theta$ 也看成是对应恒等于 1 的输入 x_0 的权值，这时式(5.1)的和式变成

$$I = \sum_{j=0}^{n} w_j x_j \tag{7.2}$$

式中，$w_0=-\theta$；$x_0=1$。

7.1.2　激发函数

常用的激发函数如图 7.2 所示。图中(a)和(b)为阈值型函数；(c)为饱和型函数；(d)是双曲型函数或称为对称的 sigmoid 函数 $f(x)=\tanh(x)=\dfrac{1-e^{-I}}{1+e^{-I}}$；(e)是 sigmoid 函数 $f(x)=\dfrac{1}{1+e^{-\beta I}}$，通常情况下 β 取 1；(f)是高斯函数 $f(x)=\exp\left[\dfrac{(1-c)^2}{b^2}\right]$，$c$ 是高斯函数中心值，c 为 0 时，函数以纵轴为对称轴，b 是高斯函数尺度因子，b 确定高斯函数的宽度。

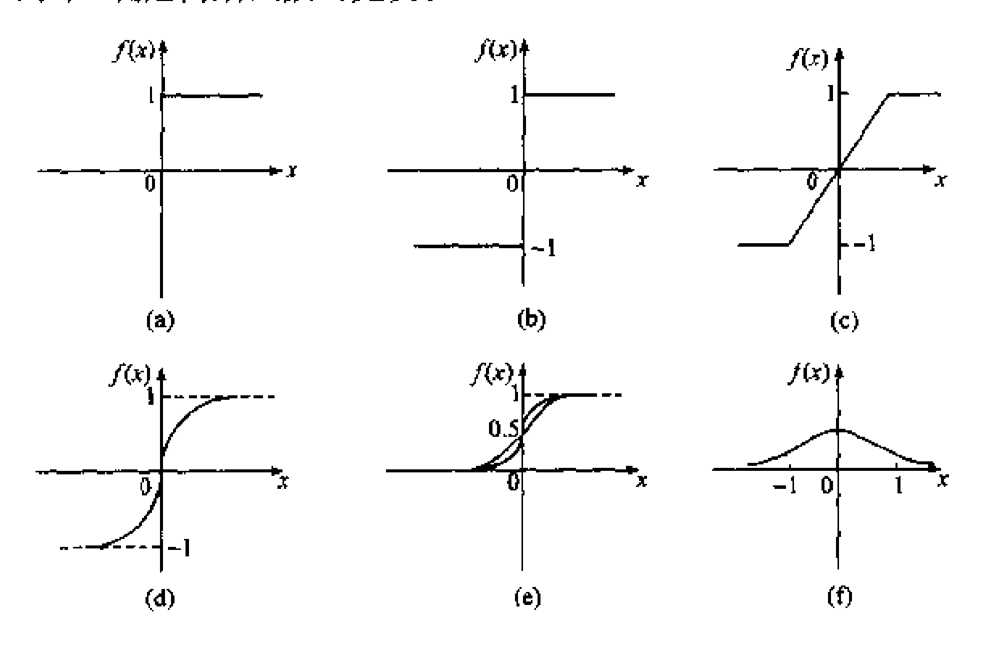


图 7.2　激发函数

7.1.3 神经网络模型分类

神经网络是由大量神经元广泛连接成的网络。根据连接方式的不同，神经网络可以分为两大类：无反馈的前向网络和相互连接型网络（包括反馈网络），如图 7.3 所示。前向网络有输入层、隐含层（简称隐层也称中间层）和输出层，隐层可以有若干层，每一层的神经元只接受前一层神经元的输出。而相互连接型网络的神经元相互之间都可能有连接，因此，输入信号要在神经元之间反复往返传递，从某一初态开始，经过若干次的变化，渐渐趋于某一稳定状态或进入周期振荡等其他状态。

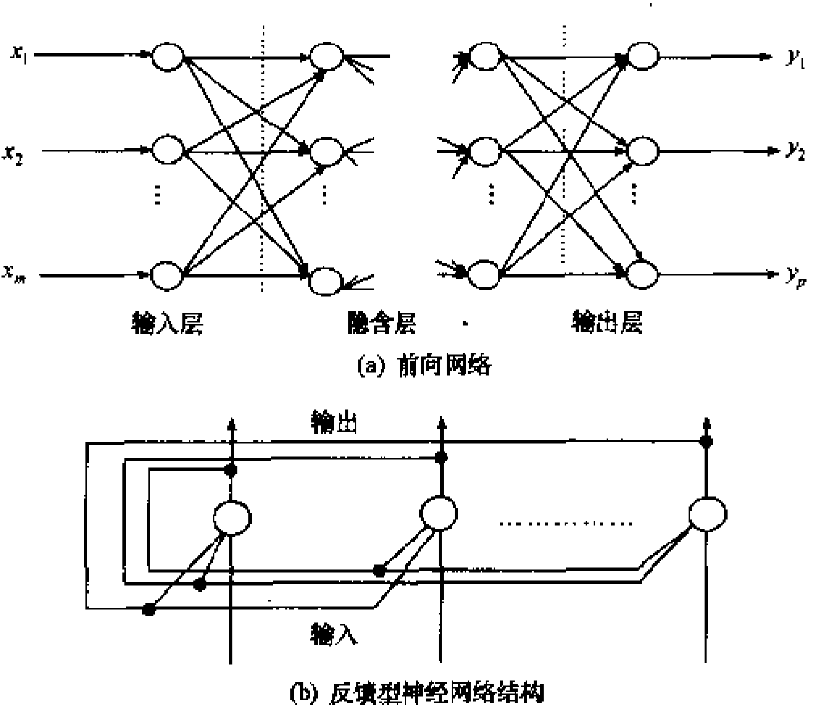


图 7.3 两种不同连接方式的网络

迄今为止，约有 40 种神经网络模型，其中有代表性的是：

具有反向传播 BP(back propagation)的网络；径向基函数 RBF(radial basis function)；GMDH(the group method of data handling)网络；感知器；CG(Cohen 和 Grossberg 提出的反馈网络模型)网络；盒中脑(BSB)模型；Hopfield 神经网络；BCM(Boltzman machine/Cauchy machine)网络；counter propagation(CPN)；Madaline 网络；自适应共振理论(ART)，它包括 ART1 和 ART2；雪崩网络；双向联想记忆(BAM)；学习矩阵(LRN)；神经认识机 ；自组织映射(SOM)；细胞神经网络(CNN)；交替投影神经网络(APNN)；小脑模型(CMAC)等。从信息传递的规律来看，这些已有的神经网络可以分成三大类：即前向网络(feedforward NN)、反馈网络(feedback NN)和自组织网络(self-organizing NN)。常用的 BP 网络是一种具

有反向传播的前向网络。

7.1.4　神经网络学习方法

1. 学习方法的种类

神经网络学习方法有多种。若按学习规则，网络的学习可分为三类：相关规则，即仅仅根据连接间的激活水平改变权值；纠错规则，即依赖于输出节点的外部反馈来改变权系数；无教师学习规则，即学习表现为自适应于输入空间的检测规则。

1) 相关规则

相关规则常用于自联想网络，执行特殊记忆状态的死记式学习，也属于无教师的学习。Hopfied 网络就是如此，所采用的是修正的 Hebb 规则。

2) 纠错规则

从方法上等效于梯度下降法，通过在局部最大改善的方向上逐步地进行修正，力图达到表示函数功能的全局解。感知器就使用纠错规则：①若一节点的输出正确，一切不变；②若输出本应为 0 而为 1，则相应的权值减小；③若输出应为 1 而为 0，则权值增加。

对于 δ 学习规则，可分为一般 δ 规则和广义 δ 规则，常见的有以下 3 种。

(1) δ 学习规则。它优于感知器学习，因为 ΔW 不是固定的量而与误差成正比，即

$$\Delta W_{ij} = \eta \delta_i V_j \tag{7.3}$$

式中，η 是全局控制系数，而 $\delta_i = t_i - V_i$，即期望值和实际值之差。

δ 学习规则和感知器学习规则一样只适用于线性可分函数，无法用于多层网络。

(2) 广义 δ 规则。它可在多网络上有效地学习，其关键是对隐节点的偏差 δ 如何定义和计算。对 BP 算法，当 i 为隐节点时，定义

$$\delta_i = f(net_i) * \sum \delta_k W_{ki} \tag{7.4}$$

式中，W_{ki}是节点 i 到上一层节点 k 的权值；$f(\cdot)$为连续的一次可微函数，将某一隐节点馈入上一层节点的误差的比例总和(加权和)作为该隐节点的误差，通过可观察到的输出节点的误差，下一层隐节点的误差就能递归得到。广义的 δ 规则可学习非线性可分函数。

(3) Boltzmann 机学习规则。它是基于模拟的统计方法来代替广义的 δ 规则。它提供了隐节点的有效学习方法，能学习复杂的非线性可分函数。其主要缺点是学习速度太慢。它基本是梯度下降法。

由此可见，纠错规则基于梯度下降法，因此不能保证得到全局最优解；同时要

求大量的训练样本,因而收缩速度慢;纠错规则对样本的表示次序变化比较敏感,这就像教师必须认真备课,精心组织才能有效地学习。

3) 无教师学习规则

在这种学习规则中,关键不在于实际节点的输出怎样与外部的期望输出相一致,而在于调整参数以反映观测事件的分布。诸如自适应共振理论(ART)、自组织特征映射和 Klopf 的享乐主义神经元都是无教师学习。

这类无教师学习的系统并不在于寻找一个特殊函数表示,而是将事件空间分类成输入活动区域,且有选择的对这些区域响应。它在应用于开发由多层竞争族组成的网络等方面有良好的前景。它的输入可以是连续值,对噪声有较强的抗干扰能力,但对较少的输入样本,结果可能依赖于输入顺序。

在人工神经网络中,学习规则是修正网络权值的一种算法,以获得合适的映射函数或其他系统性能。Hebb 学习规则的相关假设是许多规则的基础,尤其是相关规则;Hopfied 网络和自组织特征映射展示了有效的模式识别能力;纠错规则则使用梯度下降法,因而存在局部极小点问题。无教师学习提供了新的选择,它利用自适应学习方法,使节点有选择地接收输入空间上的不同特性,从而抛弃了普通神经网络学习映射函数的学习概念,并提供了基于检测特性空间的活动规律的性能描写。下面介绍几种常用的学习方法。

2. 常用神经网络的学习方法

1) Hebb 学习方法

基于对生理学和心理学的长期研究,D.O.Hebb 提出了生物神经元学,即当两个神经同时处于兴奋状态时,它们之间的连接应当加强。这一假设可描述成

$$w_{ij}(k+1) = w_{ij}(k) + I_i I_j \tag{7.5}$$

式中,$w_{ij}(k)$为连接从神经元 i 到神经元 j 的当前权值;I_i,I_j 为神经元 i,j 的激活水平。

Hebb 学习方法是一种无教师的学习方法,它只根据神经元连接间的激活水平改变权值,因此这种方法亦称为相关规则。

当神经元由式(7.1)描述时,即

$$I_i = \sum_j w_{ij} x_j - \theta_i \tag{7.6}$$

$$y_i = f(I_i) = 1/[1 + \exp(-I_i)]$$

则 Hebb 学习方法改写成

$$w_{ij}(k+1) = w_{ij}(k) + y_i y_j \tag{7.7}$$

另外,根据神经元状态变化来调整权值的 Hebb 学习方法称为微分 Hebb 学习方法,可描述为

$$w_{ij}(k+1)=w_{ij}(k)+[y_i(k)-y_i(k-1)][y_j(k)-y_j(k-1)] \tag{7.8}$$

2) 梯度下降法

梯度下降法是一种有教师信号的学习。假设下列准则函数

$$J(W)=\frac{1}{2}\varepsilon(W,k)^2=\frac{1}{2}[Y(k)-\hat{Y}(W,k)]^2 \tag{7.9}$$

式中,$Y(k)$代表希望的输出;$\hat{Y}(W,k)$为期望的实际输出;W 是所有权值组成的向量;$\varepsilon(W,k)$为$\hat{Y}(W,k)$对 $Y(k)$的偏差。现在的问题是如何调整 W 使准则函数最小。梯度下降法(gradient decline method)可用来解决此问题,其基本思想是沿着 $J(W)$的负梯度方向不断修正 $W(k)$值,直至 $J(W)$达到最小。这种方法的数学表达式为

$$W(k+1)=W(k)+\mu(k)\left(-\frac{\partial J(W)}{\partial W}\right)\bigg|_{W=W(k)} \tag{7.10}$$

式中,μ 是控制权值修正速度的变量。$J(W)$的梯度为

$$\frac{\partial J(W)}{\partial W}\bigg|_{W=W(k)}=-\varepsilon(W,k)\frac{\partial\hat{Y}(W,k)}{\partial W}\bigg|_{W=W(k)} \tag{7.11}$$

在上述问题中,把网络的输出看成是网络权值向量 W 的函数,因此网络的学习就是根据希望的输出和实际之间的误差平方最小原则来修正网络的权向量。根据不同形式的 $\hat{Y}(W,k)$,可推导出相应的算法:δ 规则和 BP 算法。

3) Deta(δ)规则

在 B.Widrow 的自适应线性元件中,自适应线性元件的输出表示为

$$\hat{Y}(W,k)=W^{\mathrm{T}}X(k) \tag{7.12}$$

式中,$W=(w_0,w_1,\cdots,w_n)^{\mathrm{T}}$ 为权值向量;$X(k)=(x_0,x_1,\cdots,x_n)^{\mathrm{T}}(k)$为 k 时刻的输入模式。

因此准则函数 $J(W)$的梯度为

$$\frac{\partial J(W)}{\partial W}\bigg|_{W=W(k)}=-\varepsilon(W,k)\frac{\partial\hat{Y}(W,k)}{\partial W}\bigg|_{W=W(k)}=-\varepsilon(W,k)X(k)\big|_{W=W(k)}$$

当 $\mu(k)=\alpha/\|X\|^2$ 时,则有 Widrow 的 δ 规则为

$$W(k+1)=W(k)+\frac{\alpha}{\|X(k)\|^2}\varepsilon(W(k),k)X(k) \tag{7.13}$$

式中,α 是控制算法和收缩性的常数,实际中往往取 $0.1<\alpha<1.0$。

为了便于计算,δ 规则可以表示成下面形式

$$W(k+1)=W(k)+\eta\varepsilon(W(k),k)X(k) \tag{7.14}$$

或

$$W(k+1) = W(k) + \eta(1-\alpha)\varepsilon(W(k),k)X(k) + \alpha(W(k) - W(k-1)) \tag{7.15}$$

式中，η 常取 $0.01 \leqslant \eta \leqslant 10.0$，$\alpha$ 取 0.9。

4) BP 算法

误差反向传播的 BP 算法(back-propagation algorithm,简称为 BP 算法)，早在 1974 年 Webos 在他的论文中提出一种 BP 学习理论，到 1985 年发展了 BP 网络训练算法。BP 网络不仅有输入层结点、输出层节点，而且有隐层节点(可以是一层或多层)。其作用函数通常选用连续可导的 Sigmoid 函数

$$f(x) = 1/[1+\exp(-x)] \tag{7.16}$$

在系统辨识中常用的是一种典型的多层并行网，即多层 BP 网。这是一种正向的、各层相互全连接的网络。对于输入信号要经过输入层，向前传递给隐层节点。经过激发(作用)函数后，把隐层节点的输出传递到输出节点，再经过激发函数后才给出输出结果。如果输出层得不到期望的输出，则转入反向传播，将误差信号沿原来的连接通路返回。通过修改各层神经网络的权值，使过程的输出 y_p 和神经网络模型的输出 y_m 之间的误差信号最小为止。BP 算法是梯度下降法的改进算法，在 7.3.2 中讨论 BP 网络时展开讨论。

5) 竞争式学习

竞争式学习属于无教师学习方式。这种学习方式是利用不同层间的神经元发生兴奋性连接，以及同一层内距离很近的神经元间发生同样的兴奋性连接，而距离较远的神经元产生抑制性连接。在这种连接机制中引入竞争机制的学习方式称为竞争式学习。其本质在于神经元网络中高层次的神经元对低层次的神经元的输入模式进行识别。竞争式机制的思想源于人的大脑的自组织能力，所以将这神经网络称为自组织神经网络[自适应共振网络模型(adaptive resonance theory,ART)]。自组织神经网络要求识别与输入最匹配的节点，定义距离 $d_j = \sum_{i=0}^{N-1}(u_i - w_{ij})^2$ 为接近距离测度，具有最短距离的节点选作胜者。它的权值向量经修正 $\Delta w_{ij} = a(u_i - w_{ij}), i \in N_c$；$\Delta w_{ij} = 0, i \notin N_c$，使该节点对输入更敏感，其中 N_c 是 N 个输入变量中距离半径较小或接近于 0 的部分。

7.1.5　神经元网络特点

神经元网络作为一种新技术迅速发展，并越来越多地用于复杂系统的辨识和控制领域，是因为与传统的辨识和控制技术相比，神经网络具有以下主要特性：

1) 非线性特性

神经元网络在理论上可以趋近任何非线性的映射。对于非线性复杂系统的建模、预测，神经元网络比其他方法更实用与经济。

2）平行分布处理

神经网络具有高度平行的结构，这使其本身可平行实现。故较其他常规方法有更大程度的容错能力。

3）硬件实现

神经网络不仅可以平行实现，而且一些制造厂家已经用专用的 VLSI 硬件来制作神经网络。

4）学习和自适应性

利用系统实际统计数据，可以对网络进行训练。受适当训练的网络有能力泛化，即当输入出现训练中未提供的数据时，网络也有能力进行辨识。神经网络还可以在线训练。

5）数据融合

网络可以同时对定性定量的数据进行操作。在这方面，网络正好是传统工程（定量数据）和人工智能领域（符号数据）信息处理技术之间的桥梁。

6）多变量系统

神经网络能处理多输入信号，且可以具有多个输出，故适用于多变量系统。从辨识和控制理论观点来看，神经网络处理非线性的能力是最有意义的。

7.2　神经网络模型辨识中常用结构

模型辨识中有正向建模和逆向建模的结构，正向建模中分为串-并辨识的结构及并联辨识结构；逆向建模又分为直接逆向辨识结构和特殊逆向辨识结构。为了在模型辨识中能选准结构，下面分析它们各自的优缺点。

系统辨识中一个重要问题是系统的可辨识性，即给定一个特殊的模型结构，被辨识的系统是否可以在该结构内适当地被表示出来。因此，必须预先给予假设（如果采用神经网络方法对系统进行辨识时），即所有被研究的系统都属于所选神经网络可以表示的一类里，根据这个假设，对同样的初始条件和任何特定输入，模型和系统应产生同样的输出。因此，辨识的系统就是根据模型系统过程的输出误差，利用网络的算法调节神经元网络的参数。直到模型参数收敛到它们的期望值。如果神经元网络训练过程要表示系统正向动态，这种建模方法就叫做正向辨识建模。其结构图如图 7.4 所示。图中 TDL 代表具有抽头的延时线，神经元网络与过程平行，系统与神经网络的预估误差用作网络的训练信号，这种结构也称为串-并辨识模型。学习的方法是有监督的，教师信号（即系统的输出）直接向神经网络提供目标值，通过网络将预估误差直接反传进行训练。假定被辨识系统具有形式

$$y_p(k+1) = f[y_p(k), y_p(k-1), \cdots, y_p(k-n+1);$$



$$u(k),u(k-1),\cdots,u(k-m+1)] \tag{7.17}$$

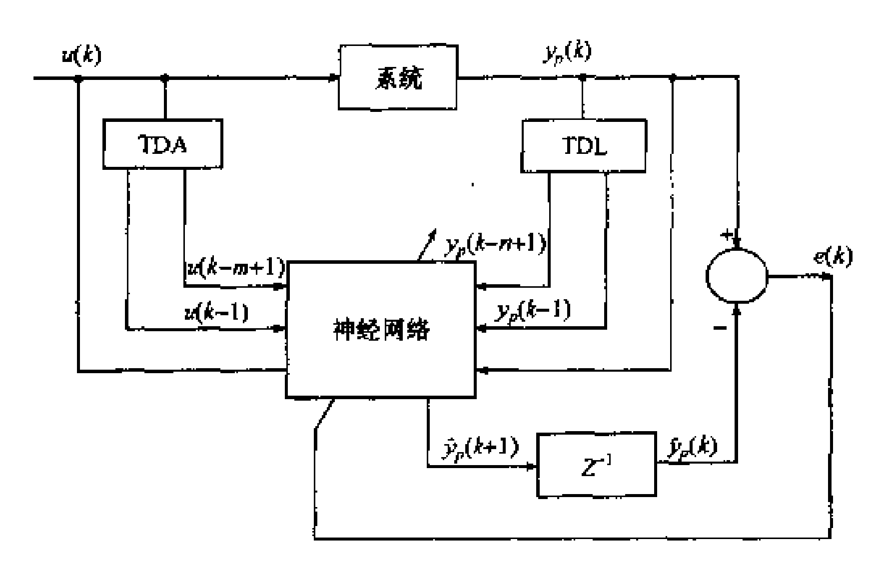


图 7.4 正向建模(串-并辨识)结构图

式(7.17)是非线性的离散时间差分方程，$y_p(k+1)$代表在时间 $k+1$ 时刻系统的输出。它是给神经网络提供的教师信号，它取决于过去 n 个输出值、过去 m 个输入值。这里只考虑系统的动态部分，还没有计入系统所承受的扰动。为了系统建模，可将神经网络的输入输出结构选择得与要建模的系统式(7.17)相同，用 y_m 表示网络的输出，则神经网络模型为

$$\begin{aligned} y_m(k+1) = \hat{y}_p(k+1) = \hat{f}[&y_p(k),y_p(k-1),\cdots,y_p(k-n+1);\\ &u(k),u(k-1),\cdots,u(k-m+1)] \end{aligned} \tag{7.18}$$

式中，$\hat{f}$表示网络输入-输出的映射。注意输入到网络的量包括了实际系统的过去值。很明显，在这个结构中利用神经网络输入，再加入系统的过去值，扩大了输入空间。但这实质上完成了基于当前最新观测数据对系统输出的一步超前预报，所以称为一步预报模型。在这种串-并联辨识模型的建立过程中，神经网络的训练等同于静态非线性函数的逼近问题。因为系统是输入有界和输出有界的，所以在辨识过程中所用的信息也是有界的。而且在模型(神经网络)中不存在反馈，因此可以保证辨识系统的稳定性。用这种方法得到的模型，通过自身输出反馈构成一个系统，也可以预测被测对象未来的输出，但只有当对象具有较强的收缩特性时，才能保证模型和系统之间的误差趋近于零。

如果将图 7.4 中神经网络模型的输入中 y_p 换成 y_m，模型的输入是由过程的输入和模型自身的输出反馈构成的，这就是正向建模的并联连接结构，如图 7.5 所示。其模型的输出可写成式(7.19)的形式。

$$y_m(k+1) = \hat{f}[y_m(k),y_m(k-1),\cdots,y_m(k-n+1);$$

$$u(k),u(k-1),\cdots,u(k-m+1)] \tag{7.19}$$

式中，$\hat{f}$ 表示网络输入-输出的映射。在这种情况下，输入到网络的量包括了神经网络模型的过去值。虽然神经网络模型内部还是静态函数，但整体上构成一个动态系统。在训练时通过对网络的参数调整，使模型动态映射到输出，从而逼近被测系统的非线性变化轨线。这种并联辨识的模型，在输入作用下，可以得到更好的输出预测。在这种意义下，我们称该模型为常时段预测模型。这种结构的模型比图 7.4 结构的模型对非线性特性有更好的逼近能力。图 7.5 是一种常用形式。

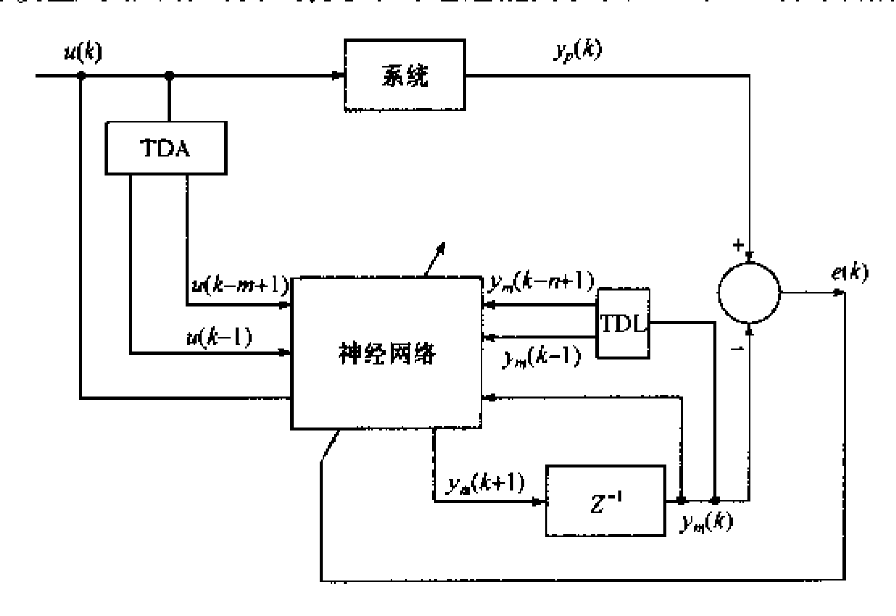


图 7.5　正向建模(并联辨识)结构图

除了上述正向建模(串-并辨识、并联辨识)结构外，还有逆向建模结构。在逆向建模中，有直接逆向辨识和特殊逆向辨识结构。直接逆向辨识建模结构如图 7.6 所示。从图可见，待辨识系统的输出作为神经网络(NN)的输入，NN 输出与系统的输入比较，用其误差来进行训练 NN，因而 NN 将通过学习建立系统的逆模型。由于这种学习不是目标导向的，在实际工作中，系统的输入 $u(k)$ 通常是经过调节器得到的，不可能预先定义，因此采用图 7.7 所示的双网辨识结构。对于被辨识系统，网络 NN_1 和 NN_2 的权值同时调节，$e_2=u-v$ 是两个网络的训练信号。当 e_1 接近 0 时，网络 NN_1 和 NN_2 将是系统逆模型的一个好的近似。这里要求 NN_1 和

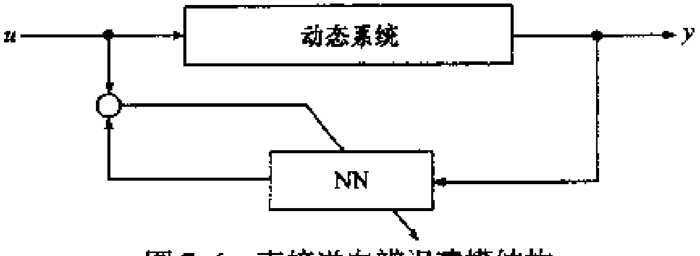


图 7.6　直接逆向辨识建模结构

NN_2 是相同结构的网络，即网络的输入层、隐层和输出层神经元节点的数目相同。当网络训练好以后，网络 NN_1 相当于一个前馈控制器，使系统的输出 $y(k)$ 和期望值接近一致，即 e_1 接近于 0，网络 NN_2 是系统的逆模型。这种双网结构又称为神经网络的直接逆控制。

另外，如果把图 7.7 所示的双网辨识结构稍作改动，即用 NN_1 的输出 $u(k)$ 作为 NN_2 的输入，NN_2 的输出和系统的输出 $y(k)$ 之差来调节 NN_2 的权值；用系统的输出 $y(k)$ 和系统输出期望值 $y_d(k)$ 之差来调整 NN_1 的权值。这样的网络结构被称为正-逆建模结构，NN_2 变为正向辨识建模网络，则 NN_1 是逆控制器。

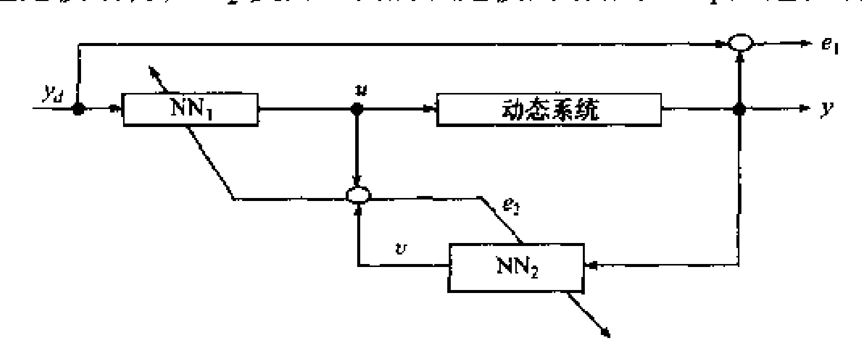


图 7.7　双网结构(直接逆控制)

7.3　辨识中常用网络训练算法

系统辨识和自适应控制是紧密相关的，从上述辨识的常用结构可以看出，正向建模中被辨识系统类似于自适应控制系统的参考模型；双网辨识结构中的系统输出期望值 $y_d(k)$ 也类似于参考模型的输出。因此，这一节首先讨论模型参考自适应系统的基本结构；其次是对在系统辨识中常用的 BP 网络训练算法进行较详细地描述。

7.3.1　自适应控制系统基本结构

自适应控制系统是一种高性能复杂控制系统。它既不同于一般的反馈控制系统，也不同于确定性最优控制系统和随机最优控制系统。要阐明自适应控制系统的概念，先要明确它与最优控制系统的区别。这种区别主要在于所研究的对象不同，要解决的问题不同。模型参考自适应系统是最常用的一种自适应系统，它由参考模型、可调系统和自适应机构三个部分组成。常见的一种结构为并联模型参考自适应系统如图 7.8 所示。图中可调系统包括受控对象、前置控制器和反馈控制器。对可调系统的工作要求，如时域指标 $\delta_p\%$、ξ、t_s 或通频带等可由参考模型直接决定。因此，参考模型实际上是一个理想的控制系统，这就是模型参考自适应系

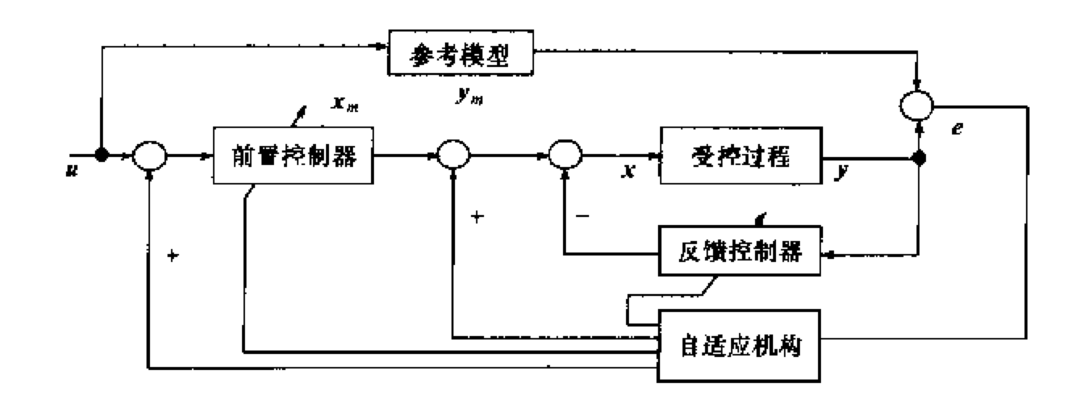


图 7.8　并联模型参考自适应系统

统不同于其他形式的控制之处,它无需进行性能指标变换。参考模型与可调系统两者性能之间的一致性由自适应机构保证,所以,自适应机构的设计十分关键,性能一致性程度由状态向量

$$e_x = x_m - x \tag{7.20}$$

或输出误差向量

$$e_y = y_m - y \tag{7.21}$$

来度量,式中,x_m,y_m和x,y分别为参考模型与受控系统的状态和输出。只要误差向量e不为零,自适应机构就按减小偏差方向修正或更新控制律,以便使系统性能指标达到或接近希望的性能指标。具体实施时,可更新前置和反馈控制器参数,也可直接改变加到输入端的信号,前者称为参数自适应方案,后者称为信号综合自适应方案。

适当变更参数模型和可调系统的相对位置,便可得到其他结构形式的模型参

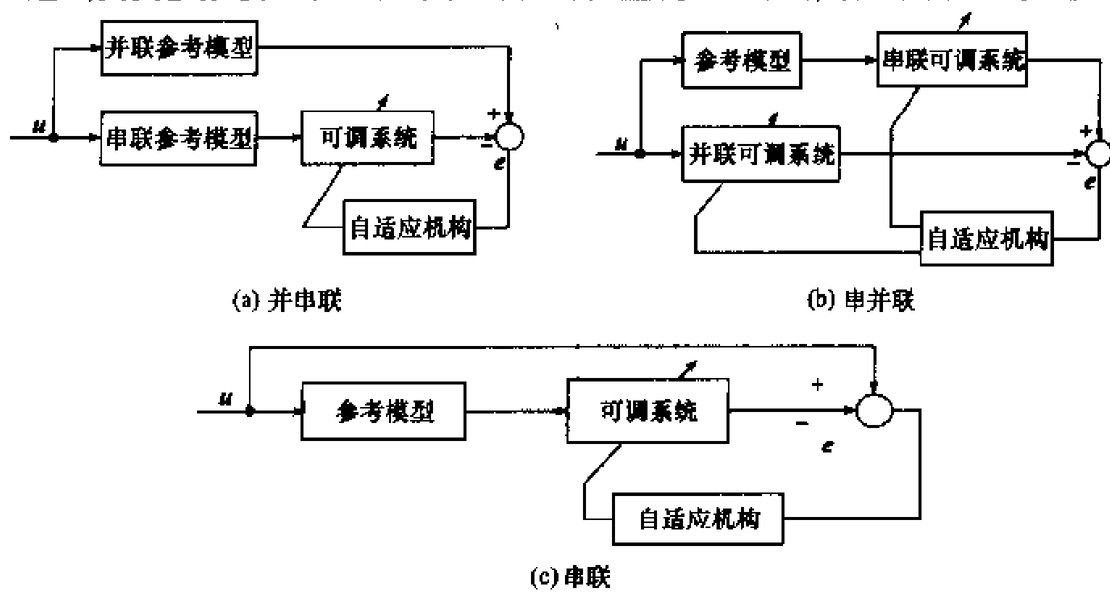


图 7.9　其他形式的模型参考自适应系统

考自适应控制系统，如图 7.9 所示。无论这些结构的形式有多大的差异，对它们进行分析与综合的方法却是基本相同的。

7.3.2 辨识中常用 BP 网络训练算法

在非线性系统模型辨识中，常用的是一种典型的多层并行网，即多层 BP 网络。其激发函数通常选用连续可导的 Sigmoid 函数

$$f(x) = 1/[1+\exp(-x)] \tag{7.22}$$

若被辨识的模型特性在正负区间变化时，则激发函数选对称的 Sigmoid 函数，又称双曲函数

$$f(x) = \tanh(x) = \frac{1-\exp(-x)}{1+\exp(-x)} \tag{7.23}$$

设三层 BP 网络如图 7.10 所示，输入层有 M 节点，输出层有 L 节点，而且隐层只有一层，具有 N 个节点。一般情况下 $N>M>L$

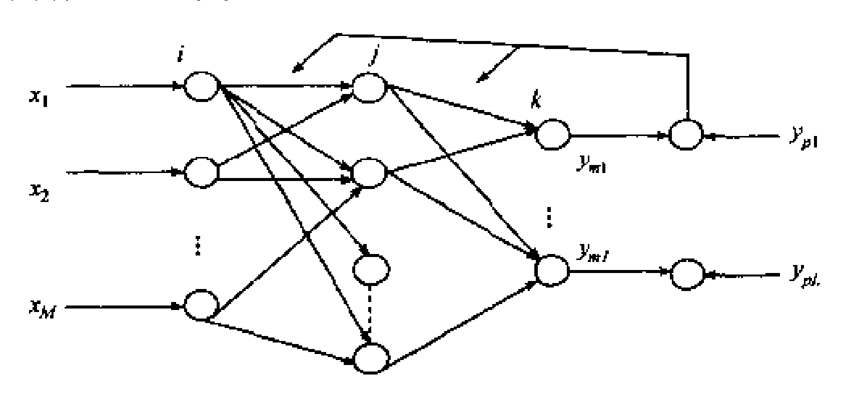


图 7.10 三层 BP 网络

设输入层神经节点的输出为 $a_i(i=1,2,\cdots,M)$；隐层节点的输出为 $a_j(j=1,2,\cdots,N)$；输出层神经节点的输出为 $y_k(k=1,2,\cdots,L)$；神经网络的输出向量为 $\boldsymbol{y}_m$；期望的网络输出向量为 $\boldsymbol{y}_p$。下面讨论一阶梯度优化方法，即 BP 算法 。

1) 网络各层神经节点的输入输出关系

输入层第 i 个节点的输入为

$$net_i = \sum_{i=1}^{M} x_i + \theta_i \tag{7.24}$$

式中，$x_i(i=1,2,\cdots,M)$为神经网络的输入；θ_i 为第 i 个节点的阈值。对应的输出为

$$a_i = f(net_i) = \frac{1}{1+\exp(-net_i)} = \frac{1}{1+\exp\left(-\sum_{i=1}^{M} x_i - \theta_i\right)} \tag{7.25}$$

隐层第 j 个节点的输入为

$$net_j = \sum_{j=1}^{N} w_{ij} a_i + \theta_j \tag{7.26}$$

式中，w_{ij}，θ_j分别为隐层的权值和第j个节点的阈值。对应的输出为

$$a_j = f(net_j) = \frac{1}{1+\exp(-net_j)} = \frac{1}{1+\exp\left(-\sum_{j=1}^{N} w_{ij} a_i - \theta_j\right)} \tag{7.27}$$

输出层第k个节点的输入为

$$net_k = \sum_{k=1}^{L} w_{jk} a_j + \theta_k \tag{7.28}$$

式中，w_{jk}，θ_k分别为输出层的权值和第k个节点的阈值。对应的输出为

$$y_k = f(net_k) = \frac{1}{1+\exp(-net_k)} = \frac{1}{1+\exp\left(-\sum_{k=1}^{L} w_{jk} a_j - \theta_k\right)} \tag{7.29}$$

2) BP网络权值调整规则

定义每一样本的输入输出模式对的二次型误差函数为

$$E_p = \frac{1}{2}\sum_{k=1}^{L} (y_{pk} - a_{pk})^2 \tag{7.30}$$

则系统的误差代价函数为

$$E = \sum_{p=1}^{P} E_p = \frac{1}{2}\sum_{p=1}^{P}\sum_{k=1}^{L} (y_{pk} - a_{pk})^2 \tag{7.31}$$

式中，P和L分别为样本模式对数和网络输出节点数。问题是如何调整连接权值使误差代价函数E最小。下面讨论基于式(7.30)的一阶梯度优化方法，即最速下降法。

(1) 当计算输出层节点时，$a_{pk} = y_k$，网络训练规则将使E在每个训练循环按梯度下降，则权系数修正公式为

$$\Delta w_{jk} = -\eta \frac{\partial E_p}{\partial w_{jk}} = -\eta \frac{\partial E}{\partial w_{jk}} \tag{7.32}$$

为了简便，式中略去了E_p的下标；式中，net_k指输出层第k个节点的输入网络；η指梯度搜索的步长，$0<\eta<1$。于是

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial net_k}\frac{\partial net_k}{\partial w_{jk}} = \frac{\partial E}{\partial net_k} a_j \tag{7.33}$$

定义输出层的反传误差信号为

$$\begin{aligned}\delta_k = -\frac{\partial E}{\partial net_k} &= -\frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial net_k} = (y_{pk} - y_k)\frac{\partial}{\partial net_k} f(net_k) \\ &= (y_{pk} - y_k) f'(net_k)\end{aligned} \tag{7.34}$$

对式(7.29)两边求导,有

$$f'(net_k) = f(net_k)[1 - f(net_k)] = y_k(1 - y_k) \tag{7.35}$$

将式(7.35)代入式(7.34),可得

$$\delta_k = y_k(1 - y_k)(y_{pk} - y_k)(k = 1,2,\cdots,L) \tag{7.36}$$

(2) 当计算隐层节点时,式(7.30)中,$a_{pk} = a_j$,则权系数修正公式为

$$\Delta w_{ij} = -\eta \frac{\partial E_p}{\partial w_{ij}} = -\eta \frac{\partial E}{\partial w_{ij}} \tag{7.37}$$

为了简便,式中略去了 E_p 的下标,于是

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial E}{\partial net_j}a_i \tag{7.38}$$

定义隐层的反传误差信号为

$$\delta_j = -\frac{\partial E}{\partial net_j} = \frac{\partial E}{\partial a_j}\frac{\partial a_j}{\partial net_j} = -\frac{\partial E}{\partial a_j}f'(net_j) \tag{7.39}$$

式中

$$-\frac{\partial E}{\partial a_j} = -\sum_{k=1}^{L}\frac{\partial E}{\partial net_k}\frac{\partial net_k}{\partial a_j} = \sum_{k=1}^{L}\left(-\frac{\partial E}{\partial net_k}\right)\frac{\partial}{\partial a_j}\sum_{j=1}^{N} w_{jk}\, a_j$$

$$= \sum_{k=1}^{L}\left(-\frac{\partial E}{\partial net_k}\right)w_{jk} = \sum_{k=1}^{L}\delta_k\, w_{jk} \tag{7.40}$$

又由于 $f'(net_j) = a_j(1 - a_j)$,所以隐层的误差反传信号为

$$\delta_j = a_j(1 - a_j)\sum_{k=1}^{L}\delta_k w_{jk} \tag{7.41}$$

为了提高学习速率,在输出层权值修正式(7.32)和隐层权值修正式(7.37)的训练规则上再加一个"势态项"(momentum term),输出层权值和隐层权值修正式为

$$\begin{cases} w_{jk}(k+1) = w_{jk}(k) + \eta_k\delta_k a_j + \alpha_k[w_{jk}(k) - w_{jk}(k-1)] & \text{输出层} \\ w_{ij}(k+1) = w_{ij}(k) + \eta_j\delta_j a_i + \alpha_j[w_{ij}(k) - w_{ij}(k-1)] & \text{隐层} \end{cases} \tag{7.42}$$

式中,η、α 均为学习速率系数,η 为各层按梯度搜索的步长,α 是各层决定过去权值的变化对目前权值变化的影响的系数,又称为记忆因子。

下面给出在系统模型辨识中BP反向传播训练的步骤:

(1) 置各层权值和阈值的初值,w_{jk},w_{ij},θ_j 为小的随机数阵;给误差代价函数 ϵ 赋值;设置循环次数 R。

(2) 提供训练用的学习资料:输入矩阵 $x_{ki}(k=1,2,\cdots,R;i=1,2,\cdots,M)$,经过系统后可得到目标输出 y_{pk},经过神经网络后可得到 y_k,对于每组 k 进行(3)~(5);

(3) 按式(7.29)计算网络输出 y_k,按式(7.27)计算隐层单元的状态 a_j;

(4) 按式(7.36)计算输出层训练误差值 δ_k,按式(7.41)计算隐层训练误差值 δ_j;

(5) 按式(7.42)修正输出层权值和隐层权值 w_{jk} 和 w_{ij};

(6) 当每次经过训练后,判断指标是否满足精度要求,即判断误差代价函数式(7.30)是否达到 $E \leqslant \varepsilon$? 若满足要求转到(7),否则再判断是否到达设定的循环次数 $k = R$? 若循环次数等于 R,转到(7),否则转到(2),重新读取一组样本,继续循环训练网络;

(7) 停止。

BP 模型把一组样本的 I/O 问题变成一个非线性的优化问题,使用了优化中最普通的梯度下降法,用迭代运算求解权系数,相应于学习记忆问题。加入隐节点使优化问题的可调参数增加,从而可得到更精确的解。如把这种神经网络看做从输入到输出的映射,则这种映射是一个高度非线性的映射。如输入节点个数为 m,输出节点个数为 L,则网络是从 $\mathbf{R}^m \to \mathbf{R}^L$ 的映射。即

$$\boldsymbol{F}:\mathbf{R}^m \to \mathbf{R}^L, \boldsymbol{Y} = F(\boldsymbol{X}) \tag{7.43}$$

式中,X,Y 分别为样本集合和输出集合。

7.4 改进的 BP 网络训练算法

虽然 BP 网络有其重要的意义,但在实际应用中存在不少问题:

(1) 学习算法的收敛速度很慢。因为 BP 算法是以梯度下降法为基础的,只具有线性收敛速度,虽通过引入"势态项"增加了一定程度的二阶信息,但对算法的性质并无根本的改变。

(2) 学习因子和记忆因子 η,α 没有一种选择的规则,若选得过大会使训练过程引起振荡,若选得过小会使训练过程更加缓慢。

(3) 网络对初始值的敏感性。同一 BP 网络不同的初值会使网络的收敛速度差异很大。若初值权值离极小点很近,则收敛速度较快;若初值权值远离极小点,则收敛速度极慢。另外,若输入初值不合适,训练起始段就会出现振荡。

(4) 网络的隐层节点个数的选择尚无理论指导,而是根据经验选取。

(5) 从数学上看 BP 算法是一个非线性的优化问题,这就不可避免地存在局部极小问题。

近年来人们对 BP 算法做了大量研究改进工作。主要包括以下几个方面:

(1) 提高学习速率的方法研究。Jacobs 等在这个研究中做了大量工作[74]。他们主要是根据学习进展情况(一般指训练误差)在训练过程中改变学习因子。采用

这种方法改进的 BP 算法，其好处是不增加额外的计算量，通过调整学习因子基本上可保证算法的收敛，但还是不能令人满意。

(2) 利用目标函数的二阶导数信息对网络训练精度的改进。Kramer 等在这方面做了大量研究。这种方法主要是利用指标函数二阶信息，即二阶导数矩阵(Hessian)或是对二阶导数矩阵的近似，这样构成其具有超线性收敛的算法。这种研究是以非线性优化理论为基础的，是将 BP 多层网的训练问题归结为一个非线性的规划问题。这样一来，优化理论中的各种优化算法(如共轭梯度法、变尺度法、Newton 法及对这些算法的改进算法)都可以用来对非线性的规划问题求解。但这种算法的应用带来一些实际问题。BP 多层网本身就是一种并行处理结构，要采用这种改进的算法，则必须将网络的权值展开构成一个权向量来进行各种向量、矩阵运算，或者构成一个矩阵近似指标函数，该阵是关系权值向量的 Hessian 阵。这对于多层网的并行处理能力有较大影响。有些研究者将二阶信息应用到某一层或者某一节点，这在一定程度上对网络训练的精度有所改进，但又使运算工作量增加，从而影响了训练速度。

以上两种改进算法为提高神经网络的训练速度和精度的研究奠定了基础。在(1)的研究中，如果再考虑如何提高网络的抗干扰措施和一些对并行网的综合管理，将会得到更加理想的效果。在(2)问题研究上，虽然二阶导数信息使计算的复杂性增加，但网络训练的有效性可提高一到两个数量级，因此，对非线性程度不太严重、并行算法要求不太高且被测系统过渡过程较慢的情况，二阶算法是可用的。

7.4.1 基于降低网络灵敏度的网络改进算法

1) 神经网络灵敏度的定义

BP 网络学习算法的收敛速度很慢的主要原因是由于 BP 网络是一种全反传式的前向网络，即只要误差反传信号存在，网络所有层的权值统统修正，从而造成了学习时间的拖延。在上述改进 BP 算法(1)提高学习速率的方法研究的基础上，若在神经网络的误差反传权值修正时增加一个协调器，该协调器将全反传式网络变成局部反传式网络，就会使网络学习速率大大提高。协调器的协调方法要靠对网络的灵敏度的分析来决定。神经网络的灵敏度是指作为辨识器或控制器的神经网络对各种变化和扰动的自学习、自调整、自适应的能力。因此，据文献[52]和[54]，对神经网络的灵敏度做如下定义。

定义 7.1　Δy_m^i 是由信号的各种变化及扰动引起的网络输出变化，$|\Delta\varepsilon|$ 为网络输入、权值变化及被辨识系统变化引起的综合误差，对第 i 样本，神经网络辨识器的灵敏度定义为

$$S_m^i(\cdot) = \frac{\mathrm{MSE}(\Delta y_m^i)}{|\Delta\varepsilon|} \tag{7.44}$$

式中，MSE(mean square error)为方差。

2）降低神经网络灵敏度的一种方法

据前分析，这里提出用于降低系统辨识中神经网络灵敏度的一种方法，用定理 7.1 来描述：

定理 7.1　在网络输入扰动、网络参数变化或被辨识系统参数变化时，根据系统在被辨识过程中的误差，动态地控制神经网络各层权值，特别是最末一级隐层到输出层的权值矩阵（即输出层的权值矩阵）修正，可以使网络的输出的均方差 $\mathrm{MSE}(\Delta y_m^i)$ 快速减小，从而使网络灵敏度 $S_m^i(\cdot)$ 降低。

证明　据文献[54]，设一个三层 BP 网络辨识器，网络的输入层、隐层（含 $b-1$ 级）、输出层节点数分别为 m、n、p，则网络输入层到第一级隐层的权值矩阵 w_{1ij} 为 $n\times m$ 维矩阵；第二级隐层的权值矩阵 w_{2ij} 为 $n\times n$ 维矩阵；依此类推，最末一级隐层的权值矩阵 $w_{(b-1)ij}$ 为 $n\times n$ 维矩阵；输出层的权值矩阵 w_{bij} 为 $p\times n$ 维矩阵；对第 j 个输入 x_j 的网络输出分量 y_{mj} 可写成

$$y_{mj}=f_j\Big(\sum_{i=1}^{n}w_{bij}\cdots f_j\Big(\sum_{i=1}^{n}w_{3ij}f_j\Big(\sum_{i=1}^{n}w_{2ij}f_j\Big(\sum_{i=1}^{m}w_{1ij}x_j+\theta_{1j}\Big)+\theta_{2j}\Big)+\theta_{3j}\Big)+\cdots+\theta_{bj}\Big) \tag{7.45}$$

由于激发函数 Sigmoid 函数连续可微，而且作为辨识器的神经网络的输入到输出共经过了 b 级 Sigmoid 函数，因此网络输出对各层权值的微分为

$$\begin{aligned}\partial y_{mj}/\partial w_{1ij}=&\,y_{mj}(1-y_{mj})\sum_{i=1}^{n}w_{bij}\,a_{(b-1)j}(1-a_{(b-1)j})\cdots\\&\times\sum_{i=1}^{n}w_{3ij}\,a_{2j}(1-a_{2j})\sum_{i=1}^{n}w_{2ij}a_{1j}(1-a_{1j})x_j\end{aligned} \tag{7.46}$$

式中，a_{1j} 为网络第一级隐层节点的输出；a_{2j} 为第二级隐层的输出；$a_{(b-1)j}$ 为最末级隐层的输出。

$$\begin{aligned}\partial y_{mj}/\partial w_{2ij}=&\,y_{mj}(1-y_{mj})\sum_{i=1}^{n}w_{bij}\,a_{(b-1)j}(1-a_{(b-1)j})\cdots\\&\times\sum_{i=1}^{n}w_{3ij}\,a_{2j}(1-a_{2j})a_{1j}\end{aligned} \tag{7.47}$$

$$\vdots$$

$$\partial y_{mj}/\partial w_{bij}=y_{mj}(1-y_{mj})a_{(b-1)j} \tag{7.48}$$

从式(7.46) 到式(7.48)可推得，网络的各层权值变化对应的网络输出变化量可写成

$$\Delta y_{mi}\approx\Big\{y_{mj}(1-y_{mi})\sum_{i=1}^{n}w_{bij}\,a_{(b-1)i}(1-a_{(b-1)i})\cdots$$

$$\times \sum_{i=1}^{n} w_{3ij} a_{2i}(1-a_{2i}) \sum_{i=1}^{n} w_{2ij} a_{1i}(1-a_{1i}) x_j \,|\, \Delta w_{1ij} \tag{7.49}$$

$$\Delta y_{mi} \approx |\, y_{mi}(1-y_{mi}) \sum_{i=1}^{n} w_{bij} a_{(b-1)i}(1-a_{(b-1)i}) \cdots$$

$$\times \sum_{i=1}^{n} w_{3kj} a_{2i}(1-a_{2i}) a_{1i} \,|\, \Delta w_{2ij} \tag{7.50}$$

$$\vdots$$

$$\Delta y_{mi} \approx |\, y_{mi}(1-y_{mi}) a_{(b-1)i} \,|\, \Delta w_{bij} \tag{7.51}$$

由于$0<y_{mi}<1, 0<a_{1i}<1, \cdots, 0<a_{(b-1)i}<1$，从式(7.49)～(7.51)可见，当权值矩阵 w_{bij}变化一个单位，网络的输出 y_{mi}的变化最大不超过 0.25；当 $w_{(b-1)ij}$变化一个单位，系统的输出 y_{mi}的变化最大不超过$(0.25)^2=0.0625$；以此类推，$w_{(b-q)ij}$变化一个单位，系统的输出 y_{mi}的变化最大不超过$(0.25)^{q+1}$；当 $q=b-1$ 时，$w_{(b-q)ij}$即为 w_{1ij}，w_{1ij}变化一个单位，系统的输出 y_{mi}的变化最大不超过$(0.25)^b$。由于 w_{1ij}经过了 b 级 Sigmoid 函数的作用，它对网络输出的影响最小，而 w_{bij}只经过了一级 Sigmoid 函数的作用，因此它对网络输出的影响最大。

从而可见，神经网络输出层的权值阵 w_{bij}的元素值的大小对网络输出影响最大，所以，当网络参数变化或系统参数变化时，动态地控制网络各层特别是输出层的权值修正，可以使网络输出的均方差 MSE(Δy_m^i)快速减小，从而使网络灵敏度$S_m^i(\cdot)$降低。

3) 基于降低网络灵敏度的 BP 网络改进算法

根据定理 7.1，基于降低网络灵敏度的 BP 网络改进算法可描述为：

(1) 在常规 BP 学习算法基础上，在网络的误差反向传播信号线上增加一个协调器，该协调器控制各层权值的修正；

(2) 当网络的综合误差$|\Delta\varepsilon|$(包括网络输出和被辨识系统的输出误差的绝对值$|e|$(又称动态的训练误差))较大时，协调器控制网络输出层的权值阵 w_{bij}增大，使网络输出迅速变化；

(3) 当训练误差为 $10\% \leqslant |e| < 20\%$时，协调器控制网络输出层的权值阵 w_{bij}减小，同时停止其他层权值的修正，使网络灵敏度 $S_m^i(\cdot)$降低，以免网络输出过冲，造成反向误差；

(4) 当训练误差$|e|<10\%$时，协调器控制只允许靠近网络输入层的第一或第二级隐层权值修正，同时停止网络输出层的权值阵 w_{bij}和其他级隐层权值的修正，网络灵敏度 $S_m^i(\cdot)$再降低，使网络输出和被辨识系统的输出误差达到允许值。

上述基于降低网络灵敏度的 BP 网络改进算法，动态地将全局反向传播式网络变成局部反传式网络，可以使网络学习速率大大提高。

7.4.2 提高一类神经网络容错性的理论和方法

1) 神经网络容错性概述

从神经网络诞生以来,人们就一直以为人工神经网络理应具有如同生物神经网络的容错力。换言之,神经网络应具有较高的可靠性,某些能力损伤后可通过学习恢复,具有容错编码的能力,具有空间上、时间上的容错力。但遗憾的是,要使人工神经网络获得良好的容错能力并不容易,还有一大片空白值得研究。对 BP 网络容错能力最早的研究是 1992 年 Emmerson 和 Damper 所做的研究。Emmerson 通过实验表明,隐层节点多的网络并不一定比隐层节点少的网络具有更强的容错力。但是,另一种称作"倍增"隐节点的方法可以提高网络的容错力。例如,针对某具体问题采用 3-4-2 三层 BP 网络进行训练,训练成功后,将网络的结构改为 3-8-2,同时将隐层的权值阵由 4×3 改成 8×3 维,其权值均是原权值的二分之一,从而保持映射关系不变。这种网络的容错能力可望得到改善,但增加隐节点,势必使计算量增大而影响训练速度,更何况隐节点不能无限制地增加。这一方法并没有从理论上予以证明[78]。与 Emmerson 几乎同时发表这方面文章的还有 Neti 和 Schneider 等,他们把对于一个已知结构的网络寻求最大容错能力的问题化为求解有约束的非线性优化问题,且从仿真过程找到了近似解,并指出一定程度上增加隐节点可使得到的解更精确,但这一方法需要复杂繁琐的计算。

Holmstron 从理论上证明了对样本对进行一定程度的噪声污染可以提高网络的泛化能力。之后 Minnix 通过实验的方法证明噪声污染也可以提高网络的容错能力。Alan Murry 和 Peter Edwards 指出,在 BP 网络训练过程中有意对权值加一扰动项(如白噪声),可以明显提高泛化能力、降低灵敏度、提高容错能力。

文献[76](1995)给出了容错神经网络的数学模型,并针对控制中的神经元网络,定义了相关的评价函数。我们将其思想引申到辨识所用的神经网络中。

定义 7.2　设某一给定的三层神经网络的输入为 $\boldsymbol{X}$,对应的输出为 $\boldsymbol{Y}$。当神经网络发生故障 f 时,神经网络的实际输出为 $\boldsymbol{Y}^f$,则

$$E_f = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{Y}^f\|^2 \tag{7.52}$$

式中,$\|\boldsymbol{Y} - \boldsymbol{Y}^f\|^2$ 表示正常时和故障情况下网络输出的差值的二范数。式(7.52)为神经网络对故障 f 的容错评价函数,它反映了在故障 f 发生的情况下,网络执行正常功能的能力。

模型 1　对于上述给定其拓扑结构的神经网络,设其训练样本集为 $T=\{(\boldsymbol{X}_t,\boldsymbol{Y}_t)\mid t=1,2,\cdots,q\}$,若存在适当的权值矩阵 $w^*=w^*(x_{ij}^*,v_{ij}^*,a_i^*,\theta_i^*)$,显然,网络权值要受到网络输入、扰动和学习因子的影响,$E_1(w)=\frac{1}{2}\sum_{j=1}^{h}\sum_{t=1}^{q}\|\boldsymbol{Y}_t-\boldsymbol{Y}_t^f\|^2$

取极小,即

$$E_1(w^*) = \min_w E_1(w) = \min \frac{1}{2}\sum_{j=1}^{h}\sum_{i=1}^{q}\|Y_t - Y_t^f\|^2 \tag{7.53}$$

则神经网络(w^*)即为容错神经网络。

模型 2 同模型 1 的设定,若使函数

$$E_1(w) = \frac{1}{2}\sum_{i=1}^{q}\|Y_t - Y_t^0\|^2 + \frac{1}{2}\sum_{j=1}^{h}\sum_{i=1}^{q}\alpha_f\|Y_t - Y_t^f\|^2 \tag{7.54}$$

取极小,即

$$E_1(w^*) = \min E_1(w) = \min\left[\frac{1}{2}\sum_{i=1}^{q}\|Y_t - Y_t^0\|^2 + \frac{1}{2}\sum_{j=1}^{h}\sum_{i=1}^{q}\alpha_f\|Y_t - Y_t^f\|^2\right] \tag{7.55}$$

则神经网络(w^*)即为容错神经网络。式中,Y_t 为教师信号; Y_t^0 为神经网络在无故障下对应输入X_t 的输出;Y_t^f 为神经网络在故障f 发生的情况下对应输入X_t^f 的实际输出,$f=1,2,\cdots,h$;α_f 为加权因子,$0<\alpha_f<1$。

文献[77]在综述了神经网络容错性的进展后指出,迄今为止所有关于提高神经网络容错性的方法都仅仅只能通过实验证明,学术界没有衡量神经网络容错力的一般标准,更没有关于对已知提高容错力方法的证明,这不能不在一定程度上阻碍了寻找更多提高容错力方法的进程。

2) 一种提高控制系统神经网络容错性的方法

前述关于神经网络容错性的方法,无论是“倍增”法还是“噪声污染”法都是提高网络容错性的有效方法,但会使神经网络训练过程的计算工作量大增,从而影响训练速度。可以证明,增加网络的层数(对 BP 网来讲,就是增加隐层的级数)可以提高网络的容错能力,但也会影响训练速度。在此基础上,文献[52]探讨了一种用于提高系统辨识中神经网络容错性的方法,用定理 7.2 来描述。

定理 7.2 在神经网络(设网络的输入、隐层(含 $b-1$ 级)、输出层节点数分别为 m,n,p)作辨识器时,当神经网络发生故障,动态控制网络权值阵的乘积结构

$$[w_{bij}w_{(b-1)ij}\cdots w_{3ij}w_{2ij}w_{1ij}] \tag{7.56}$$

使其乘积和(这里“乘积”指网络各层权值矩阵的乘积;“和”是指该乘积结果所得的矩阵所有元素之和)尽量小,是提高网络容错能力的一种方法。

证明 网络输出 y_{mj}对网络输入分量x_j 的微分为

$$\begin{aligned}\partial y_{mj}/\partial x_j &= y_{mj}(1-y_{mj})\sum_{i=1}^{n} w_{bij}\ a_{(b-1)j}(1-a_{(b-1)j})\cdots \\ &\quad\times\sum_{i=1}^{n} w_{3ij}\,a_{2j}(1-a_{2j})\sum_{i=1}^{n} w_{2ij}a_{1j}(1-a_{1j})\ w_{1ij}\end{aligned} \tag{7.57}$$

从而有

$$\Delta y_{mj} \approx \left| y_{mj}(1 - y_{mj}) \sum_{i=1}^{n} w_{bij} a_{(b-1)j}(1 - a_{(b-1)j}) \cdots \right.$$

$$\left. \times \sum_{i=1}^{n} w_{3ij} a_{2j}(1 - a_{2j}) \sum_{i=1}^{n} w_{2ij} a_{1i}(1 - a_{1j})\ w_{1ij} \right| \Delta x_j \quad (7.58)$$

从定理7.2可知,输入变化 Δx_j 经过Sigmoid函数级数越多的权值阵对网络输出的影响越小。显见,若式(7.58)的网络输入变化一个单位,则经过

$$[w_{bij}w_{(b-1)ij}\cdots w_{2ij}w_{1ij}(0.25)^b] \quad (7.59)$$

的传递才可得到网络输出最大可能的变化 Δy_{mj}。因此,网络输入 x_i 的变化对输出的影响不仅取决于与神经网络隐层的层数有关的常数 b 的大小,而且取决于权值阵的乘积的结构 $w_{bij}w_{(b-1)ij}\cdots w_{2ij}w_{1ij}$。如果这个乘积和的绝对值很小,接近零,则 x_j 的变化将几乎不会影响输出的变化。当网络发生故障时,控制 $w_{bij}w_{(b-1)ij}\cdots w_{2ij}w_{1ij}$结构使其乘积和尽量小,少一些反映在网络输出上。另外,无论故障发生在网络的哪一部位,一旦检测到故障,控制该乘积和为最小,都可抑制故障对输出的影响。所以,在像BP网络一类神经网络系统辨识中,只要动态控制网络权值阵的乘积结构$[w_{bij}w_{(b-1)ij}\cdots w_{3ij}w_{2ij}w_{1ij}]$,则可使式(7.53)

$$E_1(w^*) = \min_w E_1(w) = \min \frac{1}{2}\sum_{j=1}^{h}\sum_{t=1}^{q} \| Y_t - Y_t^f \|^2$$

取得尽可能小的值,从而可提高网络容错力。

7.4.3 提高神经网络收敛速度的一种赋初值算法

1) 网络初值概念

神经网络的初始值对网络的训练速度有很大影响。在系统辨识中,理想的初始值可以使网络模型较快地跟踪被辨识系统而收敛到其最优解,甚至还可以避免一些局部极值点的影响。但在一般情况下,人们给权值初值赋以较小的随机数。从激发函数敏感区的分布情况来看,小的权值可保证敏感区有一定的宽度;但这又使网络的输出在一定范围内变化缓慢,必须经过一段时间的训练之后才能表现出一些被测系统的特性来。如果这些随机数与被测系统的神经网络模型的最优解相差甚大,或者当样本点不在原点附近分布或被测系统的非线性特性较严重时,这种赋初值的方法会使网络的训练速度很慢,甚至只能收敛到局部极值上。

在这方面,也有学者做过研究。Wessel和Barnard于1992年以模式分类问题为背景专门讨论了用初始化的方法避免极小点的问题。他们的基本方法是:设法使隐层节点的中、小超平面穿越样本所在的区域,并且各隐层节点的方向应有尽可能大的差异。Nguyen等从多层反馈网在频域中的特性出发,提出一种初始化的方法,其目的是使初始网络能够具有逼近任意函数的结构。Denoeux等于1993年提

出一种基于典型值的初始化方法，这种方法利用样本集中的一些典型值，对输入向量做了一种变换，使得样本输入落在一个球面上，从而使网络选取合理的初始值。1995 年文献[51]针对双曲正切作用函数也提出了基于敏感区分布网络初始化的方法，目的是设法使网络处于一个良好的初始状态。

以上这些初始化方法的研究多是基于对网络的逼近的经验提出的。在以上学者研究的启发下，2001 年文献[55]提出一种赋初值的算法：这种算法针对 BP 网络中最常用的 Sigmoid 激发函数，综合考虑了网络抗干扰性和敏感区的分布情况，较全面地对多层并行网所有权值和阈值进行设计。

2) Sigmoid 函数 $f(\cdot)$ 敏感区的分布

这里针对三层(输入层、隐层和输出层)且具有多个隐层的前向网络来分析网络的敏感区的分布情况，设网络有 m 个输入节点 $x_i(i=1,2,\cdots,m)$，第一隐层有 n 个隐层节点 $H_i(i=1,2,\cdots,n)$。一个隐层节点的函数关系可表示为

$$H = f(\boldsymbol{wx}+\theta) = f(net),\quad net = \boldsymbol{wx}+\theta \tag{7.60}$$

式中，$\boldsymbol{x}$ 为该节点的输入向量；$\boldsymbol{w}$ 为权值矩阵；θ 为阈值。这里考虑激发函数为 Sigmoid 函数，激发函数 $f(\cdot)$ 的作用使得隐层节点的输出限制在(0,1)，由式(7.60)可见，隐节点的输出是对输入向量的加权和，然后经非线性饱和变换形成的，故其输出等值面是一簇平行超曲面。当某隐节点的输出值为 a 时，式(7.60)可写成

$$H = f(\boldsymbol{wx}+\theta) = \frac{1}{1+\mathrm{e}^{-(\boldsymbol{wx}+\theta)}} = a \tag{7.61}$$

则有

$$\boldsymbol{wx} = \ln\left(\frac{a}{1-a}\right) - \theta \tag{7.62}$$

给出以下定义：

定义 7.3 一个隐节点的输出值为 a 的等值超平面定义为等值超平面 P_a，等值超曲面簇 $P=\{P_a, a\in(0,1)\}$，隐层节点的起始超平面为 P_0

$$P_a = \left\{ x \in \mathbf{R}^m : \boldsymbol{h}_w x = \frac{\ln\left(\frac{a}{1-a}\right)-\theta}{\|\boldsymbol{w}\|} \right\} \tag{7.63}$$

$$P_0 = \left\{ x \in \mathbf{R}^m : \boldsymbol{h}_w x = \frac{-\theta}{\|\boldsymbol{w}\|} \right\} \tag{7.64}$$

式中，$\boldsymbol{h}_w = \boldsymbol{w}/\|\boldsymbol{w}\|$ 表示隐层节点权值向量 $\boldsymbol{w}$ 的方向上的单位向量，称为隐层节点方向。从原点出发方向为 $\boldsymbol{h}_w$ 的直线与 P_0 的交点 X_0 称为隐层节点的起始面中心点，可表示为式(7.65)，在二维情况下，P_a 的分布如图 7.11 所示。

$$X_0 = \boldsymbol{h}_w\left(\frac{-\theta}{\|\boldsymbol{w}\|}\right) \tag{7.65}$$

对任意一种激发函数，都可以用类似的方法找到其敏感区的分布。如果激发函数取双曲（对称型的 Sigmoid）函数 $f(x)=(1-e^{-x})/(1+e^{-x})$，在这种函数作用下，隐层输出限制在 $(-1,1)$ 范围内，因此敏感区以水平等值超平面 P_0 为对称中心，若一个隐节点输出为 α，则 P_0 位于等值超平面 P_a, P_{-a} 中心。

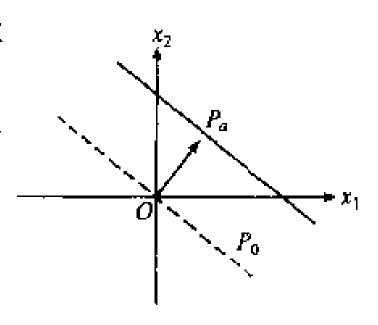


图 7.11　Sigmoid 函数作用时二维敏感区示意图

由于 Simoid 函数在 $t=0$ 附近变化最大，即函数 $f(\cdot)$ 在其值为 0.5 附近变化最明显，故其函数的导数 $f'(\cdot)$ 在 $t=0$ 即零水平等值超平面 P_0 上取极大值，所以在靠近 P_0 区域节点输出变化较明显。也就是说，靠近 P_0 区域是隐层节点映射特性的敏感区。

定义 7.4　一个隐层节点的 a 水平敏感区域 A_a 定义为节点输出在 $(0,a)$ 范围内的输入区域，可定义为

$$A_a = \{\boldsymbol{x} \in \mathbf{R}^m : 0 < f(\boldsymbol{wx}+\theta) < a\} \tag{7.66}$$

一个隐节点的 a 水平饱和区域 B_a 定义为节点输出值与饱和值相差小于 a 的输入区域，并表示为

$$B_a = \{\boldsymbol{x} \in \mathbf{R}^m : f(\boldsymbol{wx}+\theta) \geqslant a\} \tag{7.67}$$

在饱和区域 B_a 内，节点的输出几乎是不变的。

中心超曲面的位置及敏感区域的宽度是由隐层节点的权值和阈值来决定的。从式(7.62)和(7.63)可推出 a 水平敏感区域 A_a 的宽度 G_a 可表示为

$$G_a = \frac{\ln\left(\dfrac{a}{1-a}\right)}{\|\boldsymbol{w}\|} \tag{7.68}$$

从而可见，隐层节点敏感区的宽度取决于 $\|\boldsymbol{w}\|$。当 $\|\boldsymbol{w}\|$ 太大时，敏感区域非常窄，隐层节点主要工作在饱和区；当 $\|\boldsymbol{w}\|$ 太小时，敏感区域比较宽，这时隐层节点的泛化能力较强。

以上所讨论的是网络隐层的一个隐层节点的敏感区。在隐层中，每一个节点都有各自的敏感区域。这些敏感区域的延伸方向是由隐层节点的方向决定的。每个隐层节点的敏感区域的等值超平面可能是相交的，也可能是平行的，又因为每个隐层节点的敏感区域具有无限延伸的特性，这使得各隐层节点之间形成一种相互交叠相互耦合的复杂关系。这种关系给网络训练带来一定影响，因此必须协调各隐层节点的作用才能在辨识中使模型无误地跟踪被辨识系统。

3) 提高网络收敛速度的赋初值算法

考虑到网络抗干扰特性，网络输出层的权值和阈值对网络输出影响最大；同时考虑到网络激发函数的敏感区分布，这种赋初值算法[55]分两步进行：第一步是对

网络的输入到最末一级隐层之间的所有权值和阈值的设计。第二步对输出层权值和阈值进行设计。

(1) 第一步,对网络输入到最末一层隐层之间的所有权值和阈值的设计:

设被辨识系统的离散输入样本集为 $X=\{x_1,x_2,\cdots,x_q\}$,隐层节点的数目为 n,即 $H=h_j(j=1,2,\cdots,n)$,$q\gg n$,初始化的算法如下:

① 在样本集中任意取两个不同的样本点并称之为 x_{h1}、x_{h2},通过测试所辨识系统的输出得到相应的 y_{h1},y_{h2},若 $y_{h1}=y_{h2}$,则另选一个样本,最终要保证 $y_{h1}\neq y_{h2}$。

② 根据所选的样本值计算隐层节点的方向

$$\boldsymbol{h}_w = \boldsymbol{w}/\|\boldsymbol{w}\| = (\boldsymbol{x}_{h2}-\boldsymbol{x}_{h1})/\|\boldsymbol{x}_{h2}-\boldsymbol{x}_{h1}\| \tag{7.69}$$

③ 根据所选的样本值计算隐层节点敏感区的宽度

$$G_a = \frac{\ln\left(\dfrac{a}{1-a}\right)}{\|\boldsymbol{w}\|} = k_h\|\boldsymbol{x}_{h2}-\boldsymbol{x}_{h1}\| \tag{7.70}$$

式中,k_h 为[1,4]之间的随机数。

④ 计算第 i 隐层节点上的权值(隐层节点的输出 a 一般取 0.95 左右)

$$\boldsymbol{w}_i = \|\boldsymbol{w}_i\|\cdot\boldsymbol{h}_w = \frac{\ln\left(\dfrac{a}{1-a}\right)}{G_a}\cdot\boldsymbol{h}_w \tag{7.71}$$

⑤ 该隐层节点的中心为 $\boldsymbol{x}_{h0}=(\boldsymbol{x}_{h1}+\boldsymbol{x}_{h2})/2$,该隐层节点的阈值为

$$\theta_i \in -(\boldsymbol{w}_i,(\boldsymbol{x}_{h1}+\boldsymbol{x}_{h2})/2) \tag{7.72}$$

⑥ 重复①~⑤直到求出 n 对权值、阈值。即

$$\begin{cases}\boldsymbol{w}^{1h} = \boldsymbol{w}_{1ij} = [w_1,w_2,\cdots,w_n]^{\mathrm{T}}\\ \theta^{1h} = \theta_{1i} = [\theta_1,\theta_2,\cdots,\theta_n]^{\mathrm{T}}\end{cases} \tag{7.73}$$

⑦ 如果网络输入的节点数 $m>1$,这时网络的输入到第一隐层的权值阵 $\boldsymbol{w}_{1ij}$ 是 $n\times m$ 维的矩阵。①~⑥已经求出了 $\boldsymbol{w}_{1ij}$ 的第一列,其他 $m-1$ 列可以用重复求 $\boldsymbol{w}_{1ij}$ 的第一列的过程求取,但注意样本值不要重复;或者按照 $\boldsymbol{w}_{1ij}$ 的第一列的数量级随机地取值,但注意不要取相同数值。

⑧ 如果网络的隐层的级数大于 1,根据前面对网络抗干扰性的分析结果,越靠近网络输出的权值,其变化对网络的输出影响越大。因此,第二隐层的权值阵的绝对值应小于第一隐层的权值阵,即 $\boldsymbol{w}_{2ij}<\boldsymbol{w}_{1ij}$ 或 $\boldsymbol{w}^{2h}<\boldsymbol{w}^{1h}$。又考虑到隐层节点敏感区,故第二隐层的权值用下面方法求得

$$\begin{aligned}\boldsymbol{w}^{2h} = \boldsymbol{w}_{2ij} &= k_{hi}[(\boldsymbol{w}_{1ij})_1]\cdot[(\boldsymbol{w}_{1ij})_1]^{\mathrm{T}}\\ &= k_{hi}[w_1,w_2,\cdots,w_n]^{\mathrm{T}}\cdot[w_1,w_2,\cdots,w_n]\end{aligned} \tag{7.74}$$

式中，$k_{hi}\in(0.2,0.6)$，$(w_{1ij})_1$ 是第一隐层的权值阵 w_{1ij} 的第一列子阵。这样求出的 w^{2h} 是小于第一层权值阵 w_{1ij} 的 $n\times n$ 矩阵。由于 w_{1ij} 考虑了隐层节点敏感区的分布，因此，用式(7.74)求取的 w^{2h} 既保证了隐层节点敏感区有一定的宽度，又对提高网络的抗干扰性有一定的促进作用。

无论网络有多少级隐层，第 $k(k>1)$ 级隐层的权值阵都可用式(7.75)求取

$$\begin{aligned}\boldsymbol{w}^{kh}=\boldsymbol{w}_{kij}&=k_{hi}[(\boldsymbol{w}_{(k-1)ij})_1]\cdot[(\boldsymbol{w}_{(k-1)ij})_1]^{\mathrm{T}}\\&=k_{hi}[\boldsymbol{w}_{(k-1)1},\boldsymbol{w}_{(k-1)2},\cdots,\boldsymbol{w}_{(k-1)n}]^{\mathrm{T}}\cdot[\boldsymbol{w}_{(k-1)1},\boldsymbol{w}_{(k-1)2},\cdots,\boldsymbol{w}_{(k-1)n}]\end{aligned}\tag{7.75}$$

式中，$(\boldsymbol{w}_{(k-1)ij})_1$ 是第 $k-1$ 级隐层权值 $\boldsymbol{w}^{(k-1)h}$ 的第一列子阵。对应的阈值阵为

$$\boldsymbol{\theta}^{kh}=\boldsymbol{\theta}_{ki}\in-\{(\boldsymbol{w}_{kij})_{1i},(\boldsymbol{x}_{h\tau}+\boldsymbol{x}_{h(\tau+1)})/2\}\tag{7.76}$$

式中，$i=1,2,\cdots,n$；$(\boldsymbol{w}_{kij})_{1i}$ 是第 k 级隐层权值阵 w^{kh} 的第一列于阵的第 i 个元素，$(\boldsymbol{x}_{h\tau}+\boldsymbol{x}_{h(\tau+1)})/2$ 为符合初始化的算法中①的两个不同样本的中心点。

(2) 第二步，对网络输出层权值和阈值进行设计：

由于网络输出层权值 w^o 的变化对网络输出影响最大。在已知网络输入到最末一级隐层之间的权值和阈值初值的情况下(这些值第一步已全部求出)，从网络输入的变化又可以直接得到最末一级隐层的输出 $\boldsymbol{a}^{nh}$。据此，基于网络抗干扰性和最小二乘法对网络输出层 w^o 的权值进行设计，算法如下：

① 选择样本集中的特殊点：如局部极大点、局部极小点、关键性的过渡点，并求出这些样本的平均值点

$$x_s=\frac{1}{q}\sum_{i=1}^{q}x_i\tag{7.77}$$

式中，q 为特殊样本点的个数。若网络的激发函数选对称型 Sigmoid 函数，该平均值可取为零。

② 求出这些特殊点和均值点 x_s 的差值，用这些差值组成一个特殊样本集

$$\Delta X=\{\Delta x_i,i=1,2,\cdots,q\}\tag{7.78}$$

③ 当将特殊样本集中的一个变量加入到网络输入，则得到一组相应最末一级隐层的输出

$$(\boldsymbol{a}^{nh})_1=[a_{1j},a_{2j},\cdots,a_{nj}]^{\mathrm{T}}\tag{7.79}$$

如果将特殊样本集中的每一个变量都分别加入到网络，则可得到

$$\boldsymbol{a}^{nh}=\begin{bmatrix}a_{11}&a_{12}&\cdots&a_{1q}\\a_{21}&a_{22}&\cdots&a_{2q}\\\vdots&\vdots&&\vdots\\a_{n1}&a_{n2}&\cdots&a_{nq}\end{bmatrix}\tag{7.80}$$

④ 根据特殊样本集可找到对应的期望值和期望平均值的误差值，并称之为 $\boldsymbol{y}'$

$$
y^t = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1q} \\ y_{21} & y_{22} & \cdots & y_{2q} \\ \vdots & \vdots & & \vdots \\ y_{p1} & y_{p2} & \cdots & y_{pq} \end{bmatrix} \tag{7.81}
$$

式中，p，n 分别是网络输出的个数和隐层节点的个数。

⑤ 误差 y^t 可表示为 $y^t = y(1-y) \cdot a^{nh} \cdot \Delta w^o$，式中，$y$，$\Delta w^o$ 分别为网络的输出和输出层权值的变化。在最大可能情况下，$y(1-y) \approx 0.25$，$\Delta w^o \approx w^o$，从而有

$$
y(1-y) \cdot \Delta w^o \cdot a^{nh} = 0.25 \cdot w^o \cdot a^{nh} = y^t \tag{7.82}
$$

在上式中，a^{nh}，y^t 是已求出的矩阵，从而可以求出 w^o 矩阵。但 a^{nh} 不是方阵，无法求其逆阵，因此，取

$$
\boldsymbol{A} = [\boldsymbol{a}^{nh}][\boldsymbol{a}^{nh}]^{\mathrm{T}} \tag{7.83}
$$

从而得到的 $\boldsymbol{A}$ 是 $n \times n$ 阶的方阵，只要适当地调整隐层权值便可保证 $\boldsymbol{A}$ 的逆阵存在。则网络输出层的权值阵和阈值阵分别为

$$
w^o = 4y^t \cdot [\boldsymbol{a}^{nh}]^{\mathrm{T}} \cdot [\boldsymbol{A}]^{-1} \tag{7.84}
$$

$$
\boldsymbol{\theta}^o \in -\{(w^o)_1, (x_{hk} + x_{h(k+1)})/2\} \tag{7.85}
$$

式中，$(w^o)_1$ 为输出层的 $p \times n$ 维权值矩阵 w^o 的第一列子阵。

4) 应用实例

现采用这种赋初值算法对交流电动机转差角频率 $\Delta\omega$ 和定子电流 I_1 的非线性特性辨识的神经网络进行初始化，$\Delta\omega$ 和 I_1 的非线性特性描述为

$$
I_m = I_1 \sqrt{[r_2'^2 + (\Delta\omega L_2')^2] / [r_2' + (\Delta\omega(L_2' + L_m))^2]} \tag{7.86}
$$

式中 r_2'，L_2'，L_m 分别为实际运行中动态的转子等效电阻和电感，为保证磁场恒定，认为磁场电流 I_m 为定值。针对 250W 交流电动机动态实测数据，以 $\Delta\omega$ 为自变量，对应测得 I_1 特殊点上的样本集为 $X = [0.56, 0.48, 0.26, 0.17, 0.395, 0.475, 0.535]$，该样本是在额定电压 $U_e = 220$V 下，变负载，即改变和交流电动机同轴连接的直流发电机电枢上所接的可变电阻箱，转速范围 500～1420rpm，分别测得正转和反转情况下的定子电流 I_1(A)，并取其 1/3。神经网络选用“2-6-1”型 3 层网。现根据样本集，利用提高网络训练速度的初始化方法求神经网络输入到隐层之间的权值 w_{ij} 和阈值 θ_i 的初值 w_{ijc} 和阈值 θ_{ic}。从所选网络知，w_{ij} 和 θ_i 分别为 6×2 和 6×1 的矩阵。从式(7.69)求得 w_{ijc} 第一列值的方向为 $h_w = [-, -, -, +, +, +]^{\mathrm{T}}$；取 $a = 0.95$，从式(7.70)求得敏感区的宽度 $G_a = [0.32, 0.88, 0.36, 0.90, 0.32, 0.24]^{\mathrm{T}}$；从式(7.71)求得 w_{ijc} 的第一列 $w_1 = [-9.2012, -3.3459, -8.1789, 3.2716, 9.2012, 12.2683]^{\mathrm{T}}$；另取一组特殊样本值重复上述过程，并利用式(7.72)、(7.73)求得 w_{ijc} 和阈值 θ_{ic} 分别为

$$\begin{cases} w_{ijc} = [w_1, w_2] = \begin{bmatrix} -9.2012 & -3.3459 & -8.1789 & 3.2716 & 9.2012 & 12.2683 \\ -10.1324 & -5.3268 & -4.2426 & 2.8886 & 13.6686 & 2.2683 \end{bmatrix}^{\mathrm{T}} \\ \theta_{ic} = [0.8022, 0.6668, 1.02323, -0.3718, -0.9612, -1.0682]^{\mathrm{T}} \end{cases} \tag{7.87}$$

神经网络输出层的权值、阈值的初值计算过程从略。利用从电机实际运转统计的 60 组样本和计算的初值,采用 7.4.1 基于降低网络灵敏度的网络改进的 BP 算法,对交流电动机转差角频率 $\Delta\omega$ 和定子电流 I_1 的关系式(7.86)非线性特性进行辨识,结果权值 w_{ij} 和阈值 θ_i 的终值 w_{ijz} 和阈值 θ_{iz} 为

$$\begin{cases} w_{ijz} = \begin{bmatrix} 121.5773 & 137.3318 & 130.4885 & 125.5352 & -146.5570 & 137.0994 \\ 123.5773 & 134.0118 & 130.3985 & 125.5252 & -146.3570 & 137.0994 \end{bmatrix}^{\mathrm{T}} \\ \theta_{iz} = [1.1345, 0.9015, 0.9924, 1.0803, -0.7212, 0.8772]^{\mathrm{T}} \end{cases} \tag{7.88}$$

当网络的权值和阈值初值赋随机小数时

$$w_{ijc} = [0.01\ 0.01;\ 0.02\ 0.10;\ 0.11\ 0.02;\ 0.01\ 0.00;\ 0.01\ 0.10;\ 0.03\ 0.02]$$

$$\theta_{ic} = [0.30, 0.18, 0.10, 0.22, 0.12, 0.24]^{\mathrm{T}} \tag{7.89}$$

利用相同的 60 组样本和随机初值,仍采用 7.4.1 基于降低网络灵敏度的网络改进的 BP 算法对该非线性特性进行辨识,结果权值 w_{ij} 和阈值 θ_i 的终值与式(7.88)数量级几乎相同。但辨识速度明显不如前者。它们的辨识误差均方值 E 见表 7.1。

表 7.1 两种赋初值法辨识误差均方值 E 比较

方法 \ k	1	4	10	18	24	30	36	44	52	60
新赋初值法 E	0.017	0.002	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
随机初数法 E	0.115	0.005	0.008	0.003	0.002	0.002	0.001	0.001	0.001	0.001

比较新赋初值算法和随机小数初始化的辨识过程:在 P4-1.8G 的计算机上,用 MATLAB 的 m 软件编程进行辨识,前者所用的时间为 $t_s = 5.6$s,而后者时间为 $t_s = 7.5$s;从表 7.1 可知,新赋初值算法辨识误差均方值 E 比随机初值辨识的 E 也有所减小;另外,采用这种赋初值算法对多谷点、多峰值的多变量非线性函数进行辨识,由于新赋初值算法的样本包括了非线性特性的特殊点,因而可避免网络收敛于局部极小点。综上可见,新赋初值算法不仅能使网络的收敛速度有一定提高,辨识误差有所下降,且可避免局部极小点。

7.4.4 其他网络训练技巧

BP 算法在系统辨识中得到了最广泛的应用,但由于 BP 算法是基于梯度下降

法的思想,常常具有多个局部极小点。因此 BP 训练时会出现很多问题,诸如不收敛、训练速度慢或精度不高等。为了克服这些问题,除了上面讨论的几种改进算法之外,下面基于 Caudill 在 1991 年提出的 BP 网络学习窍门,加之我们在训练非线性函数和系统辨识时的体会,介绍一些 BP 训练技巧,想必对读者在神经网络系统辨识时有一定帮助。

(1) 重新给网络的权值初始化。有时由于网络的权值的初始化选得不合适,BP 算法将无法获得满意的结果,此时不妨重新设置网络权值的初值,让训练重新开始。注意权值初值不能给得太大。

(2) 给权值加扰动。在训练中,给权值加扰动,有可能使网络脱离目前局部最小点的陷阱,但仍然保持网络训练已得到的结果。如果知道网络权值的分布范围,例如在 −5 和 5 之间,则加上约 10% 的扰动,即在权值上加 −0.5 到 0.5 之间的随机数。

(3) 在网络学习的样本中适当加噪声,这是一种加快训练速度和提高抗噪声能力的行之有效的方法。在学习样本中加点噪声,可避免网络依靠死记的办法来学习,因为这种情况下所有的样本将不同(虽然有些样本相似)。

(4) 学习可允许有误差。当网络的输出和样本之间的差小于给定的误差范围,则停止对网络权值的修正。采用对网络学习宽容的做法,可加快网络的学习速度。另外还可采取自适应的办法,即允许误差在训练开始时取大点,然后随着训练逐渐减小。

(5) 网络的隐层级(层)数不能太多。网络的隐层级数尽量保持在一级,在无法达到目标时隐层可增加级数。因为在 BP 算法中,误差是通过输出层向输入层反向传播的,隐层级数越多,反向传播误差在靠近输入层时就越不可靠,这样用不可靠的误差来修正权值,其效果是可以想像的。另外隐层级数多必然计算工作量大,影响训练速度。

(6) 隐层的节点数不能太多。网络的输入层和输出层的节点个数是根据所训练函数或所辨识的实际系统的具体情况而定的。在能保证训练精度的前提下,网络隐层的节点个数一定不能太多,只要保证隐层节点数大于输入层节点数即可,否则会影响训练速度。

7.5 神经网络辨识的 MATLAB 仿真举例

7.5.1 具有噪声二阶系统辨识的 MATLAB 程序剖析

例 7.1 对具有随机噪声的二阶系统的模型辨识,进行标幺化以后系统的参考模型差分方程为

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + bu(k-1) + \xi(k) \tag{7.90}$$

式中，$a_1=0.3366$，$a_2=0.6634$，$b=0.68$，$\xi(k)$为随机噪声。由于神经网络的输出最大为 1，所以，被辨识的系统应先标幺化，这里标幺化系数为 5。利用图 7.5 正向建模(并联辨识)结构，神经网络选用 3-9-9-1 型，即输入层 i，隐层 j 包括 2 级，输出层 k 的节点个数分别为 3、9、9、1；采用 MATLAB 的 m 软件编程来说明对具有随机噪声的二阶系统的模型辨识方法；采用 7.4.1 所述的改进的 BP 算法(MBP：modified back propagation)，在常规的 BP 网络中增加协调器；激发函数采用 sigmoid 函数。

解　(附带光盘有相同程序：FLch7NNeg1.m，可直接在 MATLAB 6.L 系统运行)

1) 编程如下

```
%初始化:w10ij 表示第一隐层权值 w1ij(k-2),w11ij 表示 w1ij(k-1);w120ij 表示第二
%隐层权值 w2ij(k-2), w121ij 表示 w2ij(k-1);w20j 表示输出层权值 w3ij(k-2),w21j
%表示 w3ij(k-1);q 表示隐层阈值;p 表示输出层阈值;置标幺化系数 f1=5 等
w10ij=[.01 .01 .02;.1 .1 1 .02;.01 0 .1;.11 .01 .02;.1 .1 .02;.11 .1 .1;.1 .1 .1;0 .1 .1;.1 0 .1];
w11ij=[.1 .2 .11;.02 .13 .04;.09 .08 .08;.09 .1 .06;.1 .11 .02;.06 0 .1;.1 .1 .1;0 .10;.1 .1 .1];
w20j=[.01;.02;.1;.2;.1;.1;.1;.1;.1];   w21j=[0;0.1;.1;.02;0;.1;.1;.1;.1];
q0j=[.9 .8 .7 .6 .1 .2 .1 .1 .1];   q120j=q0j;   q11j=[.5 .2 .3 .4 .1 .2 .1 .1 .1];q12j=q11j;
w121ij=w20j*q0j;   w120ij=w20j*q11j;
f1=5;   q2j=0;   p0=.2;   k1=1;   p1=.3;   w=0;   xj=[1 1 1];   a1=[1 1 1 1];   n=100;   e1=0;
e0=0;   e2=0;   e3=0;   e4=0;   yo=0;   ya=0;   yb=0;   y0=0;   y1=0;
y2=0;   y3=0;   u=0;   u1=0;   u2=0.68;   u3=.780;   u4=u3-u2; k1=1;   kn=28;   e3=.055;
z1=0;   z12=0;   q123j=0;   t2j=0;   o12j=0; r=0; r1=0;   s=0.1;   d2j=0;
%++++++++++++++++++++++++++++++++++++++++++++++++++++%初始化结束
v1=randn(40,1);      %产生 40 个随机数,v1 代表噪声 ξ(k)
for m=1:40           % 训练 40 次开始
s1=0.1*v1(m)
yn=.3366*y2+.6634*u1+s*s1;   %参考模型 y(k)=a1y(k-1)+a2y(k-2)+bu(k-1)+ξ(k)
y1=y2;   y2=yn;   yp=yn;   % yn=y(k); y2=y(k-1); y1=y(k-2)
u0=u1;   u1=u2;   ya(m)=yn;
for k=1:n    %每个采样点训练 n 次开始
% calculating output of the hidden layer
% calculating output of the hidden layer(1)
for  i=1:9
   x1=[w11ij(i,1)*xj(:,1)]+[w11ij(i,2)*xj(:,2)]+[w11ij(i,3)*xj(:,3)];
   x=x1+q11j(:,i); o=1/[1+exp(-x)]; o11j(i)=o; end
% calculating output of the hidden layer(2)
for  i=1:9
for  j=1:9
   z1=z1+w121ij(i,j)*o11j(:,j);   end
   z=z1+q12j(:,i);   o=1/[1+exp(-x)];   o12j(i)=o;   end
```

```
% calculating output of the output layer
for  i=1:9
   yb=yb+w21j(i,:)*o12j(:,i);  end
   yi=yb+p1;  y=1/[1+exp(-yi)];
% calculating error value between aim and practice value
e0=e1;  e1=e2;  e2=[(yp-y).^2]/2;  e(k)=e2;
xj1=e2; xj2=e1; xj3=e0; xj=[xj1 xj2 xj3];
% revising right value (1)
for  i=1:9
d1=o11j(:,i)*[1-o11j(:,i)]* d2j*w21j(i,:);  %计算第1隐层误差反传信号
do=o11j(:,i)*d1;  qw=q11j(:,i)-q0j(:,i);
q2j=q11j(:,i)+.8*do+.4*qw;  q3j(:,i)=q2j;
for j=1:3
dw=w11ij(i,j)-w10ij(i,j);
w12ij=w11ij(i,j)+.8*do*xj(j)+.6*dw;
w13ij(i,j)=w12ij; end
end
w10ij=w11ij;  w11ij=w13ij;  q0j=q11j;  q11j=q3j;  %递推存储
% revising right value (2)
for  i=1:9
d1=o12j(:,i)*[1-o12j(:,i)]* d2j*w21j(i,:);  %计算第2隐层误差反传信号
do=o12j(:,i)*d1;  qw=q12j(:,i)-q120j(:,i);
t2j=q12j(:,i)+.8*do+.4*qw;  q123j(:,i)=t2j;
for  j=1:9
dw=w121ij(i,j)-w120ij(i,j);
w122ij=w121ij(i,j)+.8*do*o11j(j)+.6*dw;  w123ij(i,j)=w122ij;  end
end
w120ij=w121ij;  w121ij=w123ij;  q120j=q12j;  q12j=q123j;
% revising right value (3)
if  m<4,  r=0.2;  r1=0.0001;else,  r=0.14;  r1=0.005;  end  %协调器的作用
%  if  e2<=0.006,  r=0.0;  r1=0.0;  else break;  end
for  i=1:9
d2j=y*(1-y)*(yp-y);  %计算输出误差反传信号
dw=w21j(i,:)-w20j(i,:);
w22j=w21j(i,:)+r*d2j*o12j(i)+.4*dw+r1*e2;  w23j(i,:)=w22j;  end
w20j=w21j;  w21j=w23j;  ph=p1-p0;  p2=p1+.96*(yp-y)+.58*ph+r1*e2;
p0=p1;  p1=p2;  u=y;
if  e2<=0.005  break;  else,  end
end  %每个采样点训练n次结束
  ya(m)=yp*f1;  e3(m)=e2;  ym(m)=y*f1;  v(m)=s1;  m6=m;
end  % 训练40次结束
```

```
  w11ij = w13ij   %在 MATLAB 主界面上打印出第一隐层权值
  w121ij = w123ij   %在 MATLAB 主界面上打印出第二隐层权值
  w21j = w23j   %在 MATLAB 主界面上打印出输出层权值
m1 = m;
% grapher
subplot(3,1,1), m = 1:m6;
plot(m,ya,m,ym,'rx'), xlabel('k'), ylabel('ya and ym')
title('Identified model by MBP algorithm'),
legend('ya is system','ym is identified model'); %图标注
end
subplot(3,1,2), m = 1:m6;
plot(m,e3), xlabel('k'), ylabel('error') end
subplot(3,1,3), m = 1:m6; plot(m,v), xlabel('k'), ylabel('random noise'), end
```

2）辨识结果

辨识结果如图 7.12 所示，图中 ya 表示被辨识系统的输出，ym 表示神经网络的输出，即由改进 MBP 算法辨识的系统模型的输出。在程序中 ya 和 ym 已用标幺化系数还原到原系统值。'error'表示辨识过程中的训练误差，在 $k=3$ 以前最大误差为 2‰，之后误差在 1‰以内。'random noise'表示系统的随机噪声。各层权值如下：

w11ij =

0.2352	0.4855	0.2456
-0.1009	0.1585	0.0690
0.2093	0.1989	0.0494
0.0607	0.2364	0.1214
0.1005	0.1261	0.0211
-0.0157	-0.1511	0.0994
0.0993	0.0989	0.0994
-0.0007	0.0989	-0.1506
0.0993	0.2489	0.0994

w21j =

-0.0589
0.1011
0.0478
-0.1522
-0.1189
0.0478
0.0478
0.0478
0.0478

w121ij =

0.0165	0.0182	0.0143	0.0104	0.0023	0.0033	0.0023	0.0023	0.0023
0.0248	0.0298	0.0213	0.0129	-0.0026	-0.0008	-0.0026	-0.0026	-0.0027
0.1466	0.1672	0.1269	0.0866	0.0069	0.0168	0.0069	0.0070	0.0069
0.3043	0.3434	0.2638	0.1842	0.0238	0.0439	0.0238	0.0237	0.0238

0.1535	0.1728	0.1331	0.0934	0.0131	0.0231	0.0131	0.0130	0.0131
0.1466	0.1672	0.1269	0.0866	0.0069	0.0168	0.0069	0.0070	0.0069
0.1466	0.1672	0.1269	0.0866	0.0069.	0.0168	0.0069	0.0070	0.0069
0.1466	0.1672	0.1269	0.0866	0.0069	0.0168	0.0069	0.0070	0.0069
0.1466	0.1672	0.1269	0.0866	0.0069	0.0168	0.0069	0.0070	0.0069

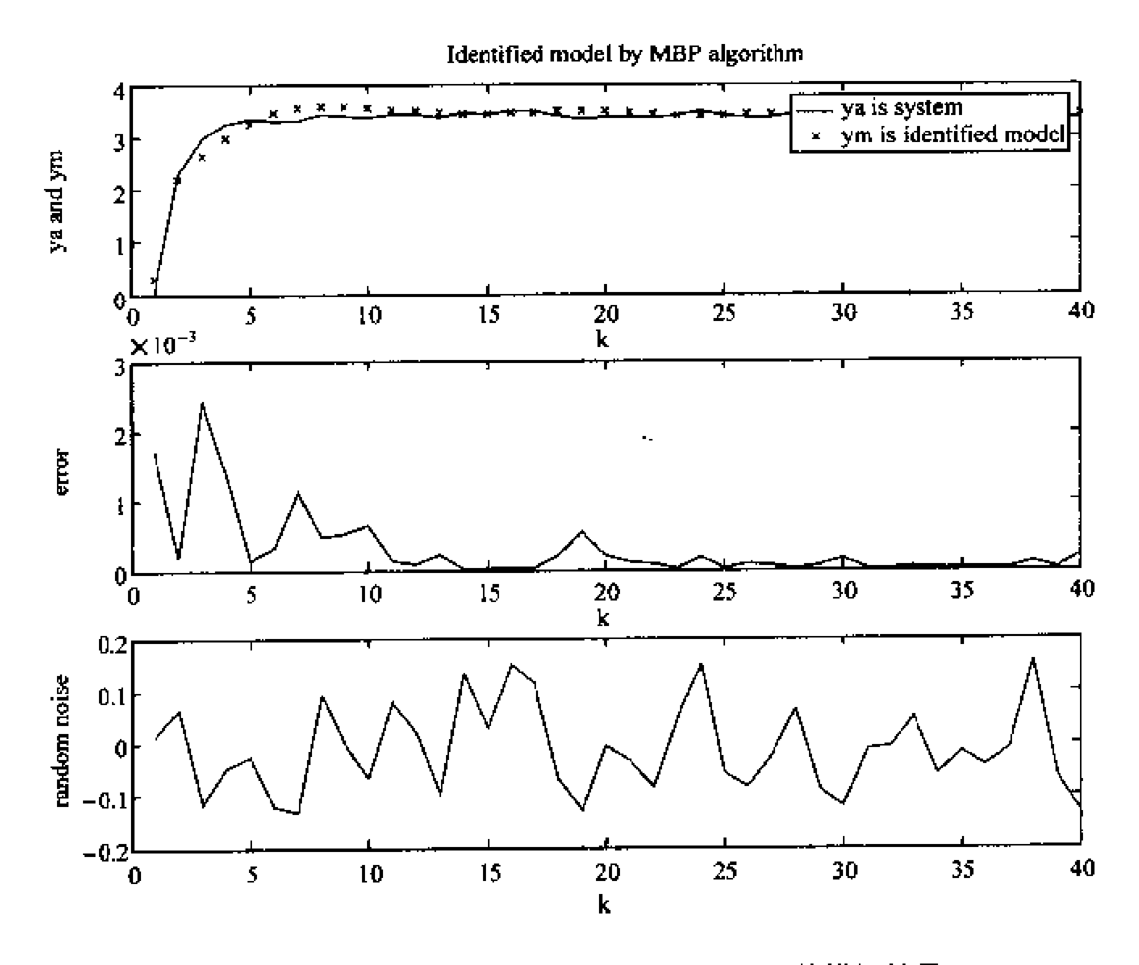


图 7.12　MBP 算法对具有随机噪声的二阶系统辨识结果

7.5.2　多维非线性辨识的 MATLAB 程序剖析

例 7.2　现以二维非线性函数 $y_p=\cos(2\pi k_1 m+2\pi k_2 n)\sin(2\pi k_2 n)$ 为样本，即 y_p 为教师信号(参考模型的输出)，其中 $0<m<36$，$0<n<36$；利用图 7.5 正向建模(并联辨识)结构，神经网络选用 3-6-1 型，即输入层 i、隐层 j、输出层 k 的节点个数分别为 3、6、1；采用 MATLAB 的 m 软件编程来说明用神经网络对多维非线性函数辨识方法。

解　由于样本幅值是正负变化的，因此，激发函数采用对称的 sigmoid 函数(双曲函数)，即

$$f(x)=\frac{1-\exp(-x)}{1+\exp(-x)}=\tanh(x) \tag{7.91}$$

采用7.4.1的改进的BP算法，即MBP算法，即由协调器来协调各层权值修正；另外，在修正权值时考虑积累误差。隐层和输出层的修正权值式为

$$w_{ij}(k+1)=w_{ij}(k)+a_1\delta_j x_i+a_2[w_{ij}(k)-w_{ij}(k-1)]+a_3 w \tag{7.92}$$

$$w_{jk}(k+1)=w_{ij}(k)+b_1\delta_k o_j+b_2[w_{ij}(k)-w_{ij}(k-1)]+b_3 w \tag{7.93}$$

式中，w 为积累误差。也可修正阈值；若不修正阈值，重新调整系数。

1）编程如下（附带光盘有相同程序：FLch7NNeg2.m，可直接在MATLAB 6.L系统运行）

```
%初始化：w10ij 表示 wij(k-2)；w11ij 表示 wij(k-1)；w20j 表示 wjk(k-2)； w21j 表示 wjk(k-1)；
% q 和 p 为阈值
w10ij=[.001 .001 .002; .001 .001 .02; .01 0 .001; .001 .001 .002; .001 0 .002; .0011 .001 .001]';
w11ij=[-.1 -.02 .11; -.21 .10 -.19; -.14 .15 -.16; .14 -.13 .17; -.13 .12 .21; -.16 -.23
.13];
w20j=[.01;.02;.1;.2;.1;.1]; w21j=[.18;.9;.9;.7;.8;.9];
q0j=[.5 .8 .4 .6 .1 .2 ]; q1j=[-.1 .02 .12 .14 -.02 .02];
q2j=0;   p0=.2;   k1=1;   p1=.2;   w23j=[0;0;0;0;0;0];   w22j=0;   w=0;   w0=0;
xj=[0.5 0.3 0.2];         % inputs
ya=[0 0 0]; yp=0; yy=0; m1=0;yam=0;yp1=0; qw=0; yo=[0 0 0]; ya1=0; error=0.0001;
n=1; q=0; e1=0; e0=0; e2=0; e3=0; e4=0; yo=0; ya=0; yb=0; y0=0; y1=0;
y2=0;   y3=0;   u=0;   u1=0;   u2=0;   k1=1;   kn=28;   e3=.055;
a1=0.036; a2=0.036; a3=0.08; w0=1; dj2=0.01;
%++++++++++++++-+++++++++++++++++++++++++++++++++++++++++ %初始化结束
for m=1:36   %变量 m 开始训练
for n=1:36   %变量 n 开始训练
  q=pi*0.05*m;   p=pi*0.02*n;
  yn=cos(2*q+2*p)*sin(2*p);   yzs=0.1*yn;   yp=yn;
for   k=1:6   %对样本的一次采样值 yp 训练 k 次开始
for   i=1:6   % calculating output of the hidden layer
  x1=[w11ij(i,1)*xj(:,1)]+[w11ij(i,2)*xj(:,2)]+[w11ij(i,3)*xj(:,3)];
  x=x1+q1j(:,i);   o=tanh(x);   o1j(i)=o;
end
% calculating output of the output layer
for   i=1:6
  yb=yb+w21j(i,:)*o1j(:,i);   end
  yi=yb+p1;   y=tanh(yi);
% calculating error value between aim and practice value
  e0=e1;   e1=e2;   e2=[(yp-y).^2]/2;
% revising right value
for   i=1:6   %修正隐层权值
```

```
   d1=[1-o1j(:,i)]*dj2*w23j(i,:);  do=o1j(:,i)*d1;  %计算隐层误差反传信号
   q3j(:,i)=q1j(i);
for  j=1:3
   dw=w11ij(i,j)-w10ij(i,j);
if  e2<0.05,  a1=0;  a2=0;  a3=0;  else,  a1=0.02;  a2=0.05;  a3=0.0005;
   end;  %协调器根据训练误差动态变化学习因子
   w12ij=w11ij(i,j)+a1*do*xj(j)+a2*dw+a3*w; w13ij(i,j)=w12ij; end
end  %修正隐层权值结束
   w10ij=w11ij;  w11ij=w13ij;  q0j=q1j;  q1j=q3j;  %递推暂存隐层权值和阈值
   w=yp-y;  w0=w;  w=0.36*w0+(yp-y);  %计算积累误差 w
if  e2<0.004, w=0.78*w; end  %协调器根据训练误差动态变化学习因子
for  i=1:6  %修正输出层权值
   d2j=y*(1-y)*(yp-y);  %计算输出层误差反传信号
   dw=w21j(i,:)-w20j(i,:);
   w22j=w21j(i,:)+0.132*d2j*o1j(i)+0.26*dw;+0.0016*w; %增加积累误差 w
   w23j(i,:)=w22j;
end
   w20j=w21j;  w21j=w23j;
   ph=p1-p0;  p2=p1+.3*(yp-y)+.102*ph;  p0=p1;  p1=p2;
if  e2<=0.005  break;  else  end  %判断误差
end  %对样本的一次采样值训练 k 次结束
   ypp(m,n)=yn;  %存储二维非线性样本
   yom(m,n)=y;  %存储神经网络辨识结果
   e3(m,n)=e2;  %存储训练误差
end  %变量 n 训练结束
   m2=m
   end  %变量 m 训练结束
   w11ij=w13ij  %句末无";",可直接在 MATLAB 主界面下观察隐层权值
   w21j=w23j  %可直接在 MATLAB 主界面下观察输出层权值
%grapher
subplot(2,2,1);
%plot3(m,n,ypp);  %也可用此语句绘制样本的三维图
mesh(ypp)  %绘制样本的三维图
xlabel('m'),ylabel('n'),zlabel('ypp'),  %三维坐标
title('Identified model by inp. algorithm'),  %图题
subplot(2,2,2); mesh(yom)
xlabel('m'),ylabel('n'),zlabel('yom');
subplot(2,1,2); mesh(e3)
xlabel('m'),ylabel('n'),zlabel('e3');
```

2) 辨识结果

对二维非线性函数 $y_p=\cos(2\pi k_1 m+2\pi k_2 n)\sin(2\pi k_2 n)$ 的辨识结果如图



7.13，图中 y_{pp} 为样本；y_{om} 为辨识结果；e_3 为训练误差。

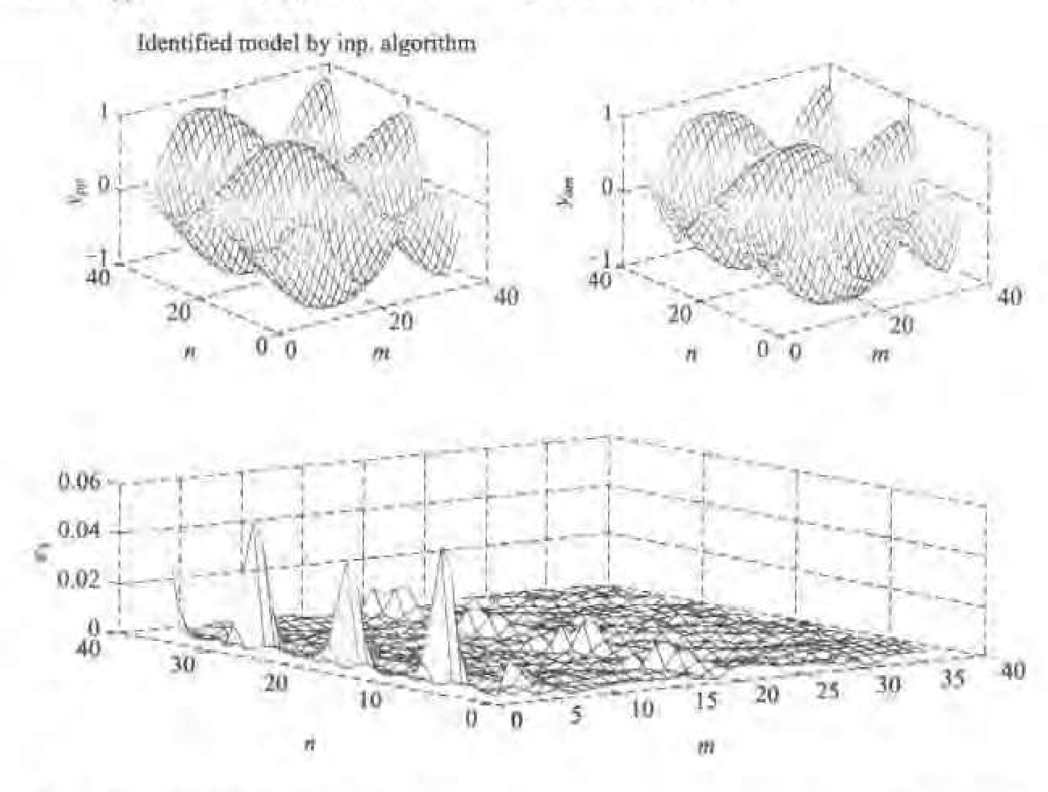


图 7.13 对二维非线性函数 $y_p=\cos(2\pi k_1 m+2\pi k_2 n)\sin(2\pi k_2 n)$ 的辨识结果

3）结论

在程序设计和调试过程中，注意以下技巧：①只要在程序中增加类似 m 或 n 的循环语句，便可用神经网对三维或三维以上的非线性函数训练，也可以用在复杂系统两个以上独立变量、任意维非线性动态参数的辨识。②程序中除了 for，end，else 和 mesh 语句外，一般应在语句末加分号。程序在 m2 = m，w11ij = w13ij 和 w21j = w23j 后不要加分号，目的有两个，一是用 m2 = m 动态地显示程序执行的状态及程序每次循环所花的时间；二是在程序运行结束显示隐层和输出层的权值：

```
w11ij =                                      w21j =
   -0.0091      0.0758      0.2083              0.8293
   -0.1206      0.1949     -0.0924              1.5274
   -0.0159      0.2657     -0.0485              0.3184
    0.2347     -0.0319      0.2698             -0.9167
   -0.0142      0.2308      0.3182              0.6995
   -0.0755     -0.1381      0.2257              1.5786
```

③调整修正因子 a1、a2、a3 和 b1、b2、b3，直到方差最小，收敛时间最短为止。④一旦修正因子、循环次数 m 和 n 确定，调整第三层内循环 k 的次数，可以明显改变训练

精度和程序运行速度，k = 20，最大方差 e3max = 0.01，程序运行时间 ts = 14s; k = 6，e3max = 0.032，ts = 12s; k = 3，e3max = 0.08，ts = 10s; k = 2，e3 = 0.22，ts = 9s. ⑤程序中，语句："if e2 < 0.05, a1 = 0; a2 = 0; a3 = 0; else, a1 = 0.02; a2 = 0.05; a3 = 0.005; end; %协调器根据训练误差动态变化学习因子"，如果在这条语句前加"%"(即不要这条语句)，训练误差将成倍增加。这种改进 BP 网络比一般的 BP 网络在 m，n，k 相同的情况下平均收敛速度约提高两秒，且训练精度明显提高。

7.6 基于改进遗传算法的神经网络及其应用

在标准的遗传算法中，基本上不用搜索空间知识或其他辅助信息，而仅用适应度函数来评估个体，并在此基础上进行遗传操作。其次，遗传算法不是采用确定性的规则，而是采用概率的变迁规则来指导其搜索方向。另外，遗传算法的处理对象不是参数本身，而是对参数集进行了编码的个体。该编码操作使得遗传算法可直接对结构对象进行操作。由于以上具有特色的操作和方法使得遗传算法使用简单易于并行化，具有较好的全局搜索性。因此遗传算法具有广泛的应用领域。但在使用遗传算法中，适应度函数条件太宽松，不易设定，常要经过多次试凑才能确定。文献[63]提出了一种改进适应度函数算法，在此基础上，将这种改进的遗传算法和神经网结合，构成了一种新型遗传神经解耦控制器。仿真和实际的炉群多变量解耦实验验证了该方法的有效性。

7.6.1 一种适应度函数的改进算法

1. 适应度函数的作用及其选择方法

适应度函数的评估是遗传算法选择、交叉、变异操作的依据。该函数不仅不受连续可微的限制，而且其定义域可以任意设定。对于适应度函数的惟一要求是对于输入目标函数有正的输出。传统的适应度函数可采用以下方法变换

$$f(x) = \begin{cases} C_{\max} - g(x), & \text{当 } g(x) < C_{\max} \\ 0, & \text{其他情况} \end{cases} \tag{7.94}$$

式中，$C_{\max}$可采用多种方式选择，$C_{\max}$可以是一个合适的输入值，也可以是在进化过程中目标函数 $g(x)$的最大值或当前群体中的最大值。从式(7.94)分析，$C_{\max}$条件太宽松，不易设定。在遗传算法中，对于目标函数 $g(x)$，若由不同的技术人员来设，结果会多种多样，差异甚大，无法判断哪种结果最合理，从而直接影响到遗传操作。

为防止遗传算法中的随机漫游现象，可通过放大相应的适应度函数的值来提高个体的竞争力。这种对适应度函数的调节称之为定标。自从 De Jong 开始引入

适应度函数定标以来,定标已成为进化过程中竞争水平的重要标志。定标有以下形式。

线性定标表示为

$$f_x(x) = af(x) + b \tag{7.95}$$

式中,$f_x(x)$为定标后的适应度函数;a 和b 可通过多种途径设置,但必须满足原适应度函数 $f(x)$是定标后的适应度函数 $f_x(x)$的平均值,或定标后的适应度函数的最大值是原适应度函数的指定倍数。除此之外,还有 σ 截断定标法、乘幂定标法等都是适应度函数的定标方法。无论以上哪种适应度函数的设定及其定标形式,均有其不确定的参数,如 C_{max}, a, b, σ 等都需要在实际遗传操作中多次摸索或试凑才能确定。特别是 C_{max},在遗传操作中无规律可寻,给遗传算法在工程中的应用带来很大不便。

2. 适应度函数的改进设计与算法

1) 适应度函数的改进设计

为了克服传统的遗传算法中适应度函数的不确定性,并且让适应度函数$f(x)$随着输入空间的个体的变化而变化,从而使适应度函数能合理地评价个体,进一步完成选择、交叉、变异的操作,本文构造以下改进适应度函数

$$f(x) = \begin{cases} C^* - g(x), & \text{当 } g(x) < C^* \\ 0, & \text{其他情况} \end{cases} \tag{7.96}$$

$$\begin{cases} C^* = \| g_m(x) - E[g(x)] \|_2 + E[g(x)] \\ g_m(x) = \max\limits_{x \in X} \{ g(x) \} \end{cases} \tag{7.97}$$

式中,g_m 是当前输入空间的个体的最大值,$E[g(x)]$是 n 个目标函数的均值。这里利用 g_m 和$E[g(x)]$之差的欧氏范数再加上 $E[g(x)]$作为 C^*,并以此取代式(7.94)中当前合适的输入值 C_{max},这样不仅能保证适应度函数 $f(x)$为非负值,且使 $f(x)$随着输入空间的个体的变化而变化。在遗传算法中,对同目标函数$g(x)$,若由不同的技术人员来设,结果会统一,从而由该适应度函数 $f(x)$来牵制遗传操作,使得遗传操作趋于更合理。

2) 适应度函数改进算法

改进适应度函数将作为遗传神经解耦控制器的一个子程序,求取子程序步骤如下:

① 从主程序读个体 x_i,计算目标函数 $g_i(x)$;

② 求 $g_i(x)$的均值,计算 $sum = sum + g_i(x)$,并求 $E[g_i(x)] = sum/i$;

③ 判断 $g_m(x) < g_i(x)$? 若是,$g_m(x) = g_i(x)$;若否,直接转④;

④ 根据式(7.97)计算

$$C^* = \| g_m(x) - E[g(x)] \|_2 + E[g(x)]$$

$$g_m(x) = \max_{x \in X} \{ g_m(x) \}$$

⑤ 根据式(7.96),求取适应度函数 $f(x)$;

⑥ 返回主程序。

7.6.2 一种改进的遗传神经解耦方法

将式(7.96)和(7.97)改进的适应度函数用于遗传操作,该遗传算法和 BP 神经网络结合,构成一种改进适应度函数的遗传神经解耦控制器。这种控制器既继承了遗传算法有“较好的全局搜索性”,又发扬了神经网络“对非线性有较强的逼近能力”的特点,因此,可以用于复杂的非线性多变量系统控制。该控制器的算法如下:①对神经网络的权值、阈值、反传误差 ϵ、循环次数 M 赋值;②按式(7.100)计算神经网络的输出:③调用改进适应度函数的遗传算法子程序,按式(7.98)和(7.99)修正神经网络权值;④判断反传误差$\leqslant\epsilon$? 若是,转⑥;⑤判断循环次数$=M$? 若是,转⑥,若否,转②;⑥结束。

改进适应度函数的遗传算法主程序中包括:①根据要解决实际问题被控量相应的动态调节系数或要逼近的函数的自变量,通过编码,产生初始群体;设定目标函数 $g(x)$;赋终结条件值。②调用改进适应度函数 $f(x)$求取子程序。③执行选择、交叉、变异遗传操作。④统计结果,群体更新;计算种群大小和遗传操作的概率式(7.99)。⑤将二进制的码译成十进制的数,输出结果解决实际问题。⑥判断终结条件到否? 若否,转②;若到,结束。

将系统的解耦控制误差作为被控量。为防止遗传算法操作中个体太长,计算量大的问题,这里采用单神经元解耦控制的方案,输入 $\boldsymbol{x}_j(i)$为三维列向量,它们分别是误差 $e(i)$、误差的变化率 $\Delta e(i)$及误差关系式 $e(i)=2e(i-1)+e(i-2)$,由遗传算法操作来完成权值矩阵 $\boldsymbol{w}(i)$的修正,$\boldsymbol{w}(i)=[w_1, w_2, w_3, \theta]^{\mathrm{T}}$,其中 w_1、w_2、w_3分别是三个输入量到单神经元的权值,θ 为单神经元的阈值,若认为权值在 $-16\sim+16$ 之间,对三个权值和阈值各用 4 位二进制码串表示,依次连接在一起形成一个应用于遗传算法的个体,该个体为 16 位的二进制码串。各连接权的字符串值和实际权值之间有如式(7.98)关系,其阈值也有相同的关系。

$$\begin{aligned} w_i(i,j) &= w_{\min}(i,j) + [\mathrm{binreplace}(i)/(2^l-1)] \\ &\quad \times [w_{\max}(i,j) - w_{\min}(i,j) + 1] \\ &= -16 + [o_i/15] \times 33 \end{aligned} \tag{7.98}$$

式中,binreplace(i)是经过遗传操作后,权值或阈值的 4 位二进制码对应的十进制数,并用 o_i 代替。目标函数选均方差值,即 $g(x)=g(e)=\dfrac{1}{2}\times[e(i)-$

$e(i-1)]^2$,适应度函数采用改进适应度函数的求取子程序计算。

在执行选择、交叉、变异遗传操作时,选择继承的概率,即评价各个权值和阈值,对应适应度函数的选择概率为

$$P_s = f(x_i)/\sum_{j=1}^{N} f_j(x) \tag{7.99}$$

式中,N 为群体中包含个体的个数,选 $N=60$,对该个体为 16 位的二进制码串进行一点交叉。在实际训练中,将选择概率较大的 20%,即前 12 个个体直接遗传给下一代。选择概率小于 0.001 的个体随机异变 16 位中的任意位。只有选择概率大于 0.001 同时处在 13 以后的个体码串进行一点交叉。这样进行遗传操作后,再利用式(7.98)将该个体为 16 位的二进制码串译成相应的权值和阈值。然后经过 sigmoid 函数激励

$$\begin{cases} y_o(i) = 1/(1+\mathrm{e}^{-net(i)}) \\ net(i) = \sum_{j=1}^{3} w_j(i)x_j(i) + \theta_i \\ u(i) = K \cdot y_o(i) \end{cases} \tag{7.100}$$

式中,$y_o(i)$是单神经元的输出;θ_i 为第 i 次遗传操作时单神经元的阈值;$u(i)$是直接加在被控对象上的输入值。

7.6.3 遗传神经解耦仿真、实验及结论

针对多变量燃烧系统的其中一个子系统,其离散模型为

$$y_i(i) = F[y_i(i-1)y_i(i-2), u(i), u(i-1)] + dy_{i\pm1}(i-1) \tag{7.101}$$

式中,$y_{i\pm1}(i-1)$为相邻的燃烧点即其他子系统对该子系统的影响,据统计,其为幅值约是 $y_i(i)$的十分之一的非线性函数[63]。针对上述多变量燃烧系统,对于两个相邻的子系统,利用 MATLAB 语言(离线仿真用 MATLAB 编程)和 C 语言结合(在实测时及控制接口用 C 编程),在 P4-1.8 PC 机上仿真,两个相邻的子系统给定温度分别为 $y_{p1}=60$℃, $y_{p2}=90$℃,先用常规的 PI 算法,并令式(7.101)的 $d=0$,仿真结果如图 7.14 所示;然后采用改进适应度函数的遗传神经解耦控制器,令式(7.101)的 $d=0.05$,仿真结果如图 7.15 所示,图中‘ - ’为给定温度;‘x’曲线为子系统 1 的输出 y_{o1};‘o’曲线为子系统 2 的输出 y_{o2}。图 7.15 下图的 wd 随机噪声曲线是模仿子系统之间耦合信号,代替相邻的燃烧点即其他子系统对该子系统的影响,分别加在各子系统中。在同样给定下,采用改进适应度函数的遗传神经解耦控制器对多变量非线性系统解耦控制,并实测温度,表 7.2 是两个实际相邻燃烧子系统实测的温度值。

以上分析、仿真及炉群实验证明,采用改进适应度函数的遗传神经解耦控制器对多变量非线性解耦控制优于传统的 PI 控制,效果良好。

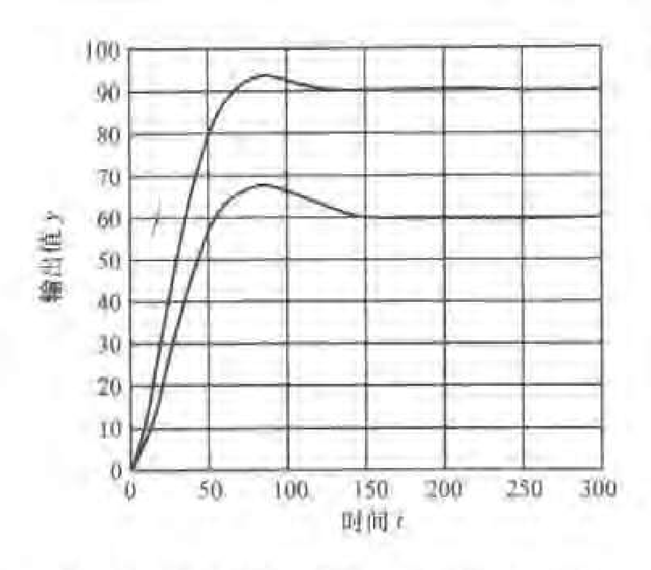


图 7.14　常规的 PID 算法解耦仿真($d=0$)时

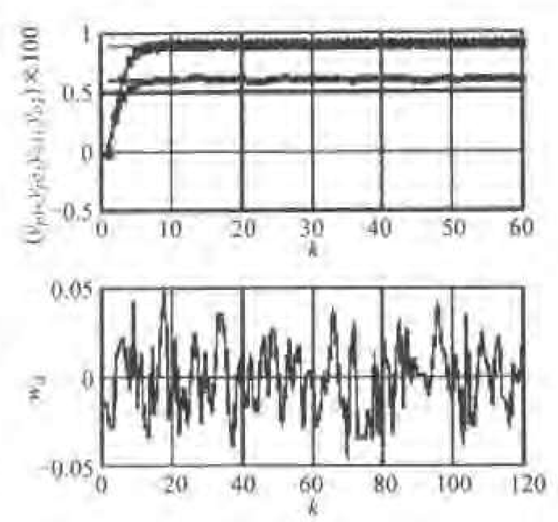


图 7.15　改进遗传神经解耦仿真($d=0.05$)

表 7.2　两个实际相邻的燃烧子系统在 60℃、90℃给定下采用改进遗传神经解耦的实测温度

时间/min	5	10	15	20	25	30	35	40	45	50	60	70	80
子系统 1 温度	14.9	26	39.1	52.4	64.6	67.1	67	66.2	64.6	62.9	62.8	62.4	62
子系统 2 温度	18.1	35.6	48.2	63.6	748.9	91.1	96.9	93.2	91.6	91.8	90.9	91.4	92

7.7　模糊神经网络及其应用

7.7.1　模糊神经网络原理及其应用

1）模糊神经网络概念

模糊神经网络(fuzzy neural network，简写 FNN)系统从结构上看主要有两类：一是在神经网络结构中引入模糊逻辑，使其具有直接处理模糊信息的能力，如果把一般神经网络中的加权求和转为模糊逻辑运算中的"∨"，"∧"("析取：即并、取大"，"合取：即交、取小")，从而形成模糊神经元网络；二是直接利用神经网络的学习功能及映射能力，去等效模糊系统中的各个模糊功能块，如模糊化、模糊推理、模糊判决。另外，还可以把神经网络和模糊控制用在同一个系统中，以发挥各自的特长。如果把 BP 算法和模糊理论结合起来就构成了上述的第一类模糊 BP 神经网络 fuzzy back propagation neural network(简称 FBP)，并将这种 FBP 用于对非线性随机函数的学习。仿真结果表明，FBP 对非线性随机函数的学习比用 BP 算法跟踪精度高。

2）FBP 学习非线性的结构框图

FBP 辨识非线性的结构框图如图 7.5 正向建模结构。从图整体来看，采用了正向建模的并联学习结构，图中 TDL 表示具有抽头的延时块。从学习非线性函数

的方法来看,采用前述局部反馈的BP算法,并在这种BP神经网络结构中引入模糊逻辑,使其具有直接处理模糊信息的能力,这里把一般BP网络中的加权求和转为模糊逻辑运算中的"∨","∧"("析取","合取"),从而形成模糊神经元网络FBP。

从图的FBP并联学习结构可知,模糊推理的结果f_z表示为

$$f_z[y_{im}(k),y_{im}(k-1),\cdots,y_{im}(k-n+1);u(k),u(k-1),\cdots,u(k-m+1),e(k)] \tag{7.102}$$

式中,y_{im}表示各层神经节点的输出。如果选三层FBP网:输入为两个节点,输入向量为$\boldsymbol{U}=[u_1,u_2]^{\mathrm{T}}$,隐层取二级并取六个神经节点,两级隐层的各神经元节点的输出分别为y_{1mj}和y_{2mj},输出取一个神经节点,其输出为y_{3m}。显然,神经网络输入到第一隐层的权值阵w_{1ij}为2×6阶矩阵,第二隐层的权值阵w_{2ij}为6×6阶矩阵,输出层的权值阵w_{3ij}为6×1阶矩阵。这种FBP具体算法如下(输入量和各权值先经过模糊化处理):

(1) 求第一隐层各神经节点的输出

$$\boldsymbol{X}=\boldsymbol{w}_{1ij}\circ\boldsymbol{U}=\begin{bmatrix} w_{111} & w_{112}\\ w_{121} & w_{122}\\ w_{131} & w_{132}\\ w_{141} & w_{142}\\ w_{151} & w_{152}\\ w_{161} & w_{162}\end{bmatrix}\circ\begin{bmatrix}u_1\\u_2\end{bmatrix}=\begin{bmatrix} w_{111}\wedge u_1\vee w_{112}\wedge u_2\\ w_{121}\wedge u_1\vee w_{122}\wedge u_2\\ w_{131}\wedge u_1\vee w_{132}\wedge u_2\\ w_{141}\wedge u_1\vee w_{142}\wedge u_2\\ w_{151}\wedge u_1\vee w_{152}\wedge u_2\\ w_{161}\wedge u_1\vee w_{162}\wedge u_2\end{bmatrix}=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\end{bmatrix} \tag{7.103}$$

式中,$w_{1ij}\circ\boldsymbol{U}$表示权值矩阵和输入向量模糊乘法运算。

$$y_{1mj}=\frac{1}{1+\mathrm{e}^{-qx_j}},\quad j=1,2,\cdots,6 \tag{7.104}$$

式中,q是将模糊量x_j转化为模拟量(即解模糊化)所加的系数,在调试程序中进行摸索,其大小和各层权值初始值有关。由于该网络的各层权值阵的初始值是根据7.4.3提高神经网络收敛速度的一种赋初值算法的方法设置的,据此,式(7.104)的q调到5.6396,式(7.105)的q调到6.4778,式(7.106)的q调到0.5163。

(2) 同理可得第二隐层各神经节点的输出

$$y_{2mj}=\frac{1}{1+\mathrm{e}^{-q[w_{2ij}\circ y_{1mj}]}},\quad i=1,2,\cdots,6,\quad j=1,2,\cdots,6 \tag{7.105}$$

(3) 神经网络的输出层输出

$$y_{3m}=\frac{1}{1+\mathrm{e}^{-q[w_{3ij}\circ y_{2mj}]}},\quad i=1,2,\cdots,6,\quad j=1,2,\cdots,6 \tag{7.106}$$

(4) 根据前述降低一类神经网络灵敏度的理论和方法的研究,在改变网络权

值时，针对学习误差的大小分别对各层权值进行修正，将误差全反传的 BP 网络变成局部反传的网络。

3) 验证及结论

用 FBP 算法和 BP 算法分别对非线性随机函数式(7.107)进行学习验证

$$y_m = a_1u(k) + d_1\sin(2\pi k)/a_2 + b_1\cos(2\pi k)/b_2 + \xi(k) \quad (7.107)$$

式中，系数 a_1, a_2, b_1, b_2 均随着学习次数 k 的变化而随机变化；$\xi(k)$为噪声。用 MATLAB 编程。BP 算法学习结果如图 7.16 所示，FBP 算法的学习结果如图 7.17 所示。

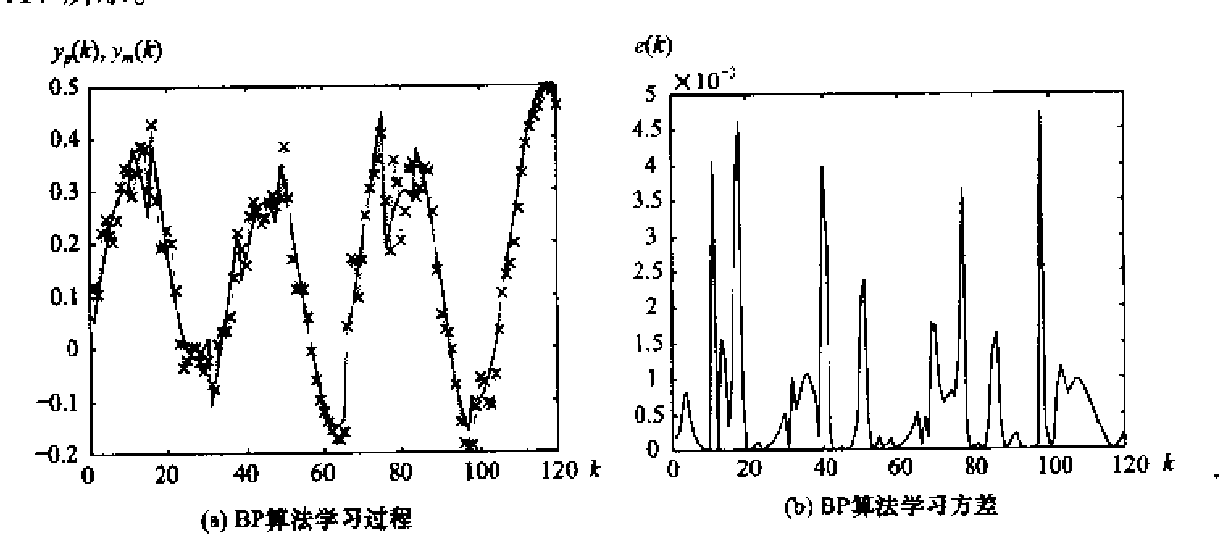


图 7.16 BP 算法学习结果

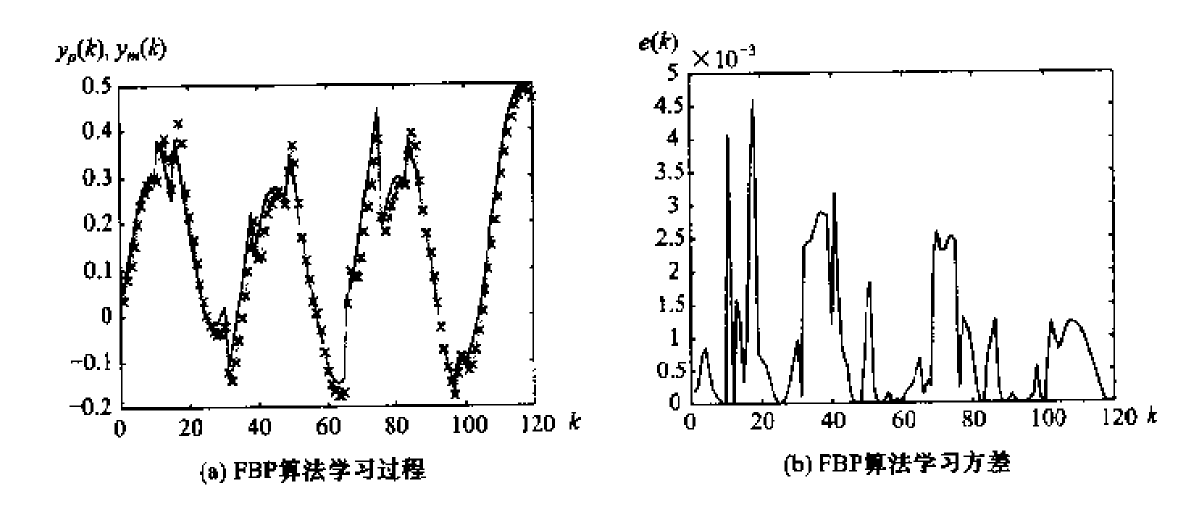


图 7.17 FBP 算法学习结果

图 7.16 和 7.17 中，横轴为学习的次数 k，图(a)中实线为 $y_p(k)$，带"×"的线为网络输出 $y_m(k)$。另外，在学习过程中给输入上叠加 0.1 幅值的噪声 $\xi(k)$，用 $\xi(k)$的 20 个离散值取均值几乎为 0，说明我们所用的噪声 $\xi(k)$是白噪声，因此辨

识的置信度为95%。

对FBP的理论分析表明:神经网络中的加权求和转为模糊逻辑运算中的"V","Λ",可以使FBP具有直接处理模糊信息的能力,同时避免了加权求和中较大数字的复杂运算。在FBP中针对学习误差的大小分别对各层权值进行修正,全反传的BP网络变成局部反传FBP,从而,不仅使网络跟踪教师信号的能力加强,而且使学习的速度比一般BP算法有所提高。图7.16和7.17的仿真结果进一步证明了这一点。

7.7.2 FNN对非线性多变量系统的MATLAB解耦仿真

1) DMFFCNN的基本模型

针对燃烧系统的非线性多变量耦合特性,文献[65]提出了一种隶属函数型神经网与模糊控制融合的解耦方法(DMFFCNN),图7.18为一种隶属函数型模糊神经网(DMFFCNN)基本模型[50],该网络的激发函数为正态形函数

$$f(x) = \exp\left[\frac{-(x-a)^2}{b^2}\right] \tag{7.108}$$

在图中,w_s表示网络中(A)层到(B)层的连接权值,即清晰量x_j转化为模糊量的量化因子;网络权值w_c表示正态函数的中心值($w_c = a$);网络权值w_d表示隶属函数$\mu(x_j)$的分布参数,又称尺度因子($b = 1/w_d$);$f(x)$表示激发(作用)函数。所以,隶属函数型模糊神经网模型的输入输出映射关系为

$$\mu_{A_i}(x_i) = \exp\left[\frac{-(w_s x_i - w_c)^2}{(1/w_d)^2}\right] \tag{7.109}$$

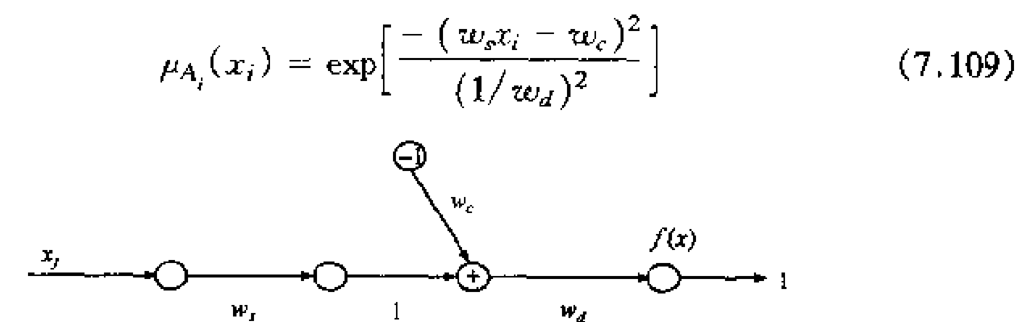


图7.18 隶属函数型模糊神经网模型

2) DMFFCNN的系统结构与方法

DMFFCNN系统结构中有两个子系统相互耦合,两个子系统的输出y_1和y_2相互耦合影响,y_{d1}和y_1是子系统1的输入给定和输出;图7.19较详细地给出了系统2的DMFFCNN框图。其中K_1、K_2为子系统2解耦过程输出和给定的误差E及其变化率的比例因子,K_3为DMFFCNN的输出和实际系统的控制量之间的比例因子。子系统图7.19中的虚线框DMFFCNN结构为神经网,其中网络为具有双输入单输出、九个隐层节点的三层网络。第一层:输入层。第二层:语言项层,

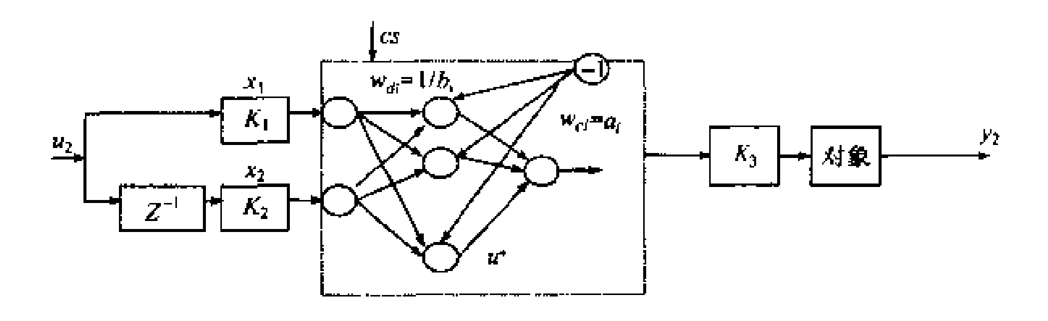


图 7.19　DMFFCNN 系统的结构图

该层使用的隶属函数类似图 7.18。cs 为网络控制算法反传的输出。用正态形函数作激发函数，根据式(7.103)及(7.105)，与输入 x_i 相关的第 j 个节点的输出为

$$y_{ij}(x_1,x_2)=\exp\left[-\frac{1}{2}\frac{(x_1-a_{1i})^2}{b_{1i}}\right]\exp\left[-\frac{1}{2}\frac{(x_2-a_{2j})^2}{b_{2j}}\right] \tag{7.110}$$

第三层：输出层，先由规则表求出 u^*

$$\text{IF}\quad x_{1i}\ \text{is}\ A_i\ \text{and}\ x_{2j}\ \text{is}\ B_j\ \text{THEN}\ u^*\ \text{is}\ C_j \tag{7.111}$$

i 表示 x_1 有 i 个语言项节点，j 表示 x_2 有 j 个语言项节点，a_{1i}、a_{2j} 表示隶属函数的中心位置，b_{1i}、b_{2j} 表示隶属函数的形状参数。输出层将前件与结论连接起来，得到数值型输出为

$$U(x_1,x_2)=\sum_{i\in I}\sum_{j\in J}u^*y_{ij}(x_1,x_2) \tag{7.112}$$

u^* 表示隐层到输出层的权值，即由模糊控制所得到的结论值。DMFFCNN 智能解耦控制器作用到子系统的离散控制量为

$$u(k)=u(k-1)+K_3\cdot U(x_1,x_2) \tag{7.113}$$

DMFFCNN 的学习方法

$$u(k+1)=u(k)+m\left(-\frac{\partial J}{\partial u}\right)+\lambda\Delta u(k) \tag{7.114}$$

式中，m 为学习率；λ 为动量因子。DMFFCNN 的学习过程是通过调整网络权值 u^* 和参数 a_1、b_1、a_2、b_2，使被控过程的输出值逼近期望输出，因此定义误差目标函数为

$$J=\frac{1}{2}(y_d-y)^2 \tag{7.115}$$

式中，y_d 为给定的期望值；y 为被控过程的输出。网络参数的调整利用最速梯度下降法，学习过程中各参数计算如下

$$\frac{\partial J}{\partial a_{1i}}=-K_3(y_d-y)\frac{\partial y(k)}{\partial u(k)}\frac{\partial U}{\partial a_{1i}}=-K_3\delta\frac{\partial U}{\partial a_{1i}}$$

$$= -K_3\delta\sum_{j\in J}u^* y_{ij}(x_1,x_2)\left[\frac{x_1(k)-a_{1i}}{b_{1i}^2}\right],\quad \delta=(y_d-y)\frac{\partial y(k)}{\partial u(k)} \tag{7.116}$$

$$\frac{\partial J}{\partial b_{1i}}=-K_3\delta\frac{\partial U}{\partial b_{1i}}=-K_3\delta\sum_{j\in J}u^* y_{ij}(x_1,x_2)\left[\frac{(x_1(k)-a_{1i})^2}{b_{1i}^3}\right] \tag{7.117}$$

$$\frac{\partial J}{\partial a_{2j}}=-K_3\delta\frac{\partial U}{\partial a_{2j}}=-K_3\delta\sum_{i\in I}u^* y_{ij}(x_1,x_2)\left[\frac{x_2(k)-a_{2j}}{b_{2j}^2}\right] \tag{7.118}$$

$$\frac{\partial J}{\partial b_{2i}}=-K_3\delta\frac{\partial U}{\partial b_{2i}}=-K_3\delta\sum_{j\in J}u^* y_{ij}(x_1,x_2)\left[\frac{(x_1(k)-a_{2i})^2}{b_{2i}^3}\right] \tag{7.119}$$

$$\frac{\partial J}{\partial u^*}=-K_3\delta\frac{\partial U}{\partial u^*}=-K_3\delta\exp\left[-\frac{1}{2}\left(\frac{x_1-a_{1i}}{b_{1i}}\right)^2-\frac{1}{2}\left(\frac{x_2-a_{2j}}{b_{2j}}\right)^2\right] \tag{7.120}$$

综合式(7.112)～(7.115)，再考虑式(7.116)～(7.120)中 u^* 和参数 a_1、b_1、a_2、b_2的变化，便可得到 DMFFCNN 的动态学习算式

$$u(k+1)=u(k)+m\left\{-K_3\delta\max\left[\frac{\partial J}{\partial a_{1i}},\frac{\partial J}{\partial b_{1i}},\frac{\partial J}{\partial a_{2j}},\frac{\partial J}{\partial b_{2j}},\frac{\partial J}{\partial u^*}\right]\right\}+\lambda\Delta u(k) \tag{7.121}$$

在动态学习中，根据式(7.121)进行差分运算，直到目标函数达到设定极小值，从式(7.121)可见，每次学习只取目标函数 J 对上述五个变量求偏导其中最大的一个，修正权值也只修正对应的一个变量。这种学习方法比同样层数和节点的全局反馈的 BP 网络程序复杂，但由于是有针对性的修正权值，计算工作量并未增加，因此，训练速度有所提高。

例 7.3　具有耦合的两个相邻子系统的差分方程为

$$\begin{cases} y_1(k)=a_{11}y(k-1)+b_{11}u(k-1)+b_{12}u(k-2)+\xi_1(k)+y_{12}(k) \\ y_2(k)=a_{21}y(k-1)+b_{21}u(k-1)+b_{22}u(k-2)+\xi_2(k)+y_{21}(k) \end{cases} \tag{7.122}$$

式中，$\xi_i(k)$随机噪声；$y_{ij}(k)$为两个相邻子系统之间的耦合；采用隶属函数型神经网与模糊控制融合的解耦方法(DMFFCNN)实现解耦控制。

解　分析：NN 作控制器和 NN 作辨识器的主要区别是，前者将 NN 的输出作用给被控系统，系统的输出 y 和理想输出 y_d比较，后者将 NN 的输出 y_m和教师信号 y_d比较，两者均根据比较的差值修正 NN 的权值。

将给定系数代入到式(7.122)，编写的程序：FLch7FNNeg3.m，程序太大有 620 多行不便写出，附在光盘中并加有注释。在程序中，两个相邻子系统的差分方程写成

y2=0.5*y1+2.5*u2+2.5*u1+n1(:,k)+0.01*y12

y22=0.5*y12+1.25*u22+1.25*u12+n(:,k)+0.01*y1

式中,n1(:,k)和 n(:,k)分别为用 n=0.28*rand(size(time)); n1=0.3*rand(size(time));产生的两个子系统的随机噪声。程序运行结果如图 7.20 所示。

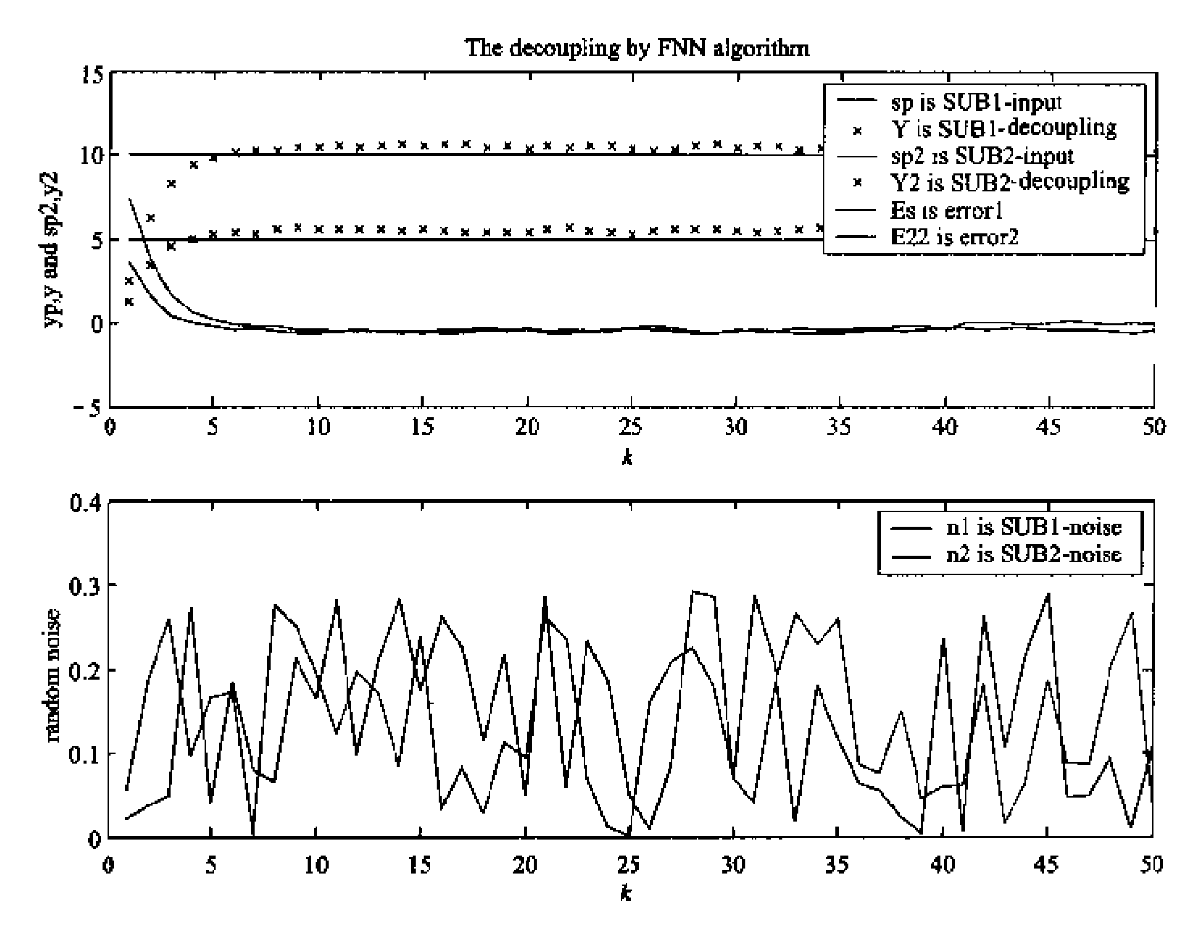


图 7.20　模糊神经网络 FNN 对相邻耦合子系统的解耦结果

两个子系统初始化,给相同隶属函数型 FNN 的中心值、尺度因子和权值阵赋值如下:

a102=[-2 0 2]; a112=[-2 0 2]; a202=[-2 0 2]; a212=[-2 0 2]; %FNN 隐层的权值阵,即中心值, a1i 和 a2i 是分别对应两个输入 x1 和 x2 的隶属函数的中心值;

b102=[1.5 1.5 1.5]; b112=[1.5 1.5 1.5]; b202=[1.5 1.5 1.5]; b212=[1.5 1.5 1.5]; %尺度因子, b1i 和 b2i 是分别对应两个输入 x1 和 x2 的隶属函数的尺度因子;

v02=[-1 -0.5 -0.5; -0.5 0 0.5;0.5 0.5 1]; v12=[-1 -0.5 -0.5; -0.5 0 0.5;0.5 0.5 1]; %FNN 输出层权值阵,v02 表示 $w_2(k-1)$;v12 表示 $w_2(k)$;

程序运行结果 FNN 模型的参数:

子系统 1 运行结果 FNN 的中心值、尺度因子和权值阵:

a11 = −5.4905 0.0016 −0.0039; a21 = −0.1085 −0.0010 −0.0001
b11 = 0.9576 5.4020 4.3817; b21 = 5.4539 7.0620 3.6550
v3 =
−1.7100 −1.2100 −1.2100
−1.2100 −0.7100 −0.2100
−0.2100 −0.2100 0.2900

子系统 2 运行结果 FNN 的中心值、尺度因子和权值阵：

a112 = −3.7450 −0.0031 0.2719; a212 = −2.5444 0.0007 0.2126
b112 = 0.2495 1.4999 3.7133; b212 = −0.1431 1.5000 3.4497
v12 =
−1.5411 −1.0411 −1.0411
−1.0411 −0.5411 −0.0411
−0.0411 −0.0411 0.4589

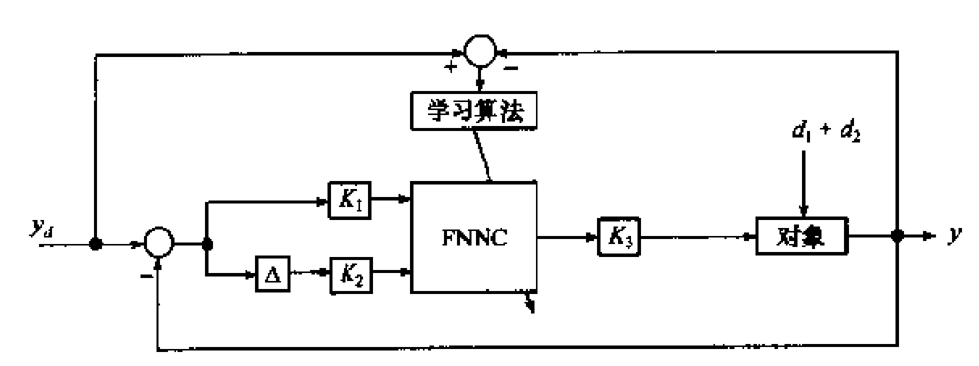
7.8 小 结

本章主要探讨神经网络辨识的理论、方法及其神经网络辨识 MATLAB 仿真。在 7.1～7.3 节中，介绍了神经网络辨识的基本概念，其中包括神经网络的模型分类、激发函数、学习方法及特点、神经网络模型辨识中常用结构、辨识中常用 BP 网络训练算法等。BP 算法在系统辨识中得到了最广泛的应用，但由于 BP 算法是基于梯度下降法的思想，常常具有多个局部极小点。因此 BP 训练时会出现很多问题，诸如不收敛、训练速度慢或精度不高等。为了克服这些问题，7.4 节探讨了改进的 BP 网络训练算法，其中包括基于降低网络灵敏度的网络改进算法、提高一类神经网络容错性的理论和方法、提高神经网络收敛速度的一种赋初值算法及其他网络训练技巧。7.5 节是神经网络辨识的 MATLAB 仿真举例，其中包括对开发的具有噪声二阶系统辨识的 MATLAB 程序的剖析、对多维非线性辨识的 MATLAB 程序的剖析，前者采用 7.4.1 所述的改进的 BP 算法(MBP)，在常规的 BP 网络中增加协调器，激发函数采用 sigmoid 函数；后者由于样本幅值是正负变化的，因此，激发函数采用对称的 sigmoid 函数或称双曲函数，采用改进的 BP 算法，即由协调器来协调各层权值修正；另外，在修正权值时考虑积累误差。7.6、7.7 节分别讨论基于改进遗传算法的神经网络和模糊神经网络及其应用，并列举了遗传神经解耦仿真及实验结果、FNN 对非线性多变量系统的 MATLAB 解耦仿真编程方法及其开发的程序。

习　题

1. 生物神经元模型的结构功能是什么?
2. 人工神经元模型的特点是什么?
3. 人工神经网络的特点是什么? 如何分类?
4. 有哪几种常用的神经网络学习算法?
5. BP算法的特点是什么?
6. 增大权值是否能够使BP学习变慢?
7. 系统参考模型为 $y(k)=-1.2y(k-1)+0.7y(k-2)+u(k-1)+0.5u(k-2)+0.1\xi(k)$, $\xi(k)$为幅值小于1的随机噪声,试分别采用3-6-1型BP神经网络和改进型的MBP网络辨识系统的模型,编写系统模型辨识程序,打印出程序运行结果,并分析采用常规BP算法和MBP算法辨识该系统模型的收敛速度和辨识精度。
8. 模糊神经网络控制系统原理结构如题8图所示,图中 y_d 为系统期望输出(单位阶跃信号),y 为系统输出,被控对象的传递函数为

$$G(s)=\frac{K_1}{T_1s+1}e^{-\tau s}=\frac{22}{1600s+1}e^{-480s}$$



题8图　模糊神经网络控制系统仿真结构示意图

设计如题8图所示结构的模糊神经网络控制器FNNC。FNNC模型为

$$R_i:\ \text{IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ THEN } \mu \text{ is } B^i,\ i=1,2,\cdots,7$$

式中,$A_1^i=$[NB,NM,NS,ZE,PS,PM,PB],隶属函数为高斯函数;

$A_2^i=$[NB,NM,NS,ZE,PS,PM,PB],隶属函数为高斯函数。

各变量论域:$X_1=[-E,+E]=[-3,+3]$, $X_2=[-\dot{E},+\dot{E}]=[-1.5,+1.5]$

$$U=[-u,+u]=[-0.5,+0.5],\ K_1=w_{s1}=\frac{n}{e}=\frac{6}{3}=2$$

$$K_2=w_{s2}=\frac{n}{\Delta e}=\frac{6}{1.5}=4,\ K_3=K_n=\frac{u}{n}=\frac{0.5}{6}$$

初值：$w_c(0)=\{-6,-4,-2,0,2,4,6\}$，$w_d(0)=\{1\}$；$w_{s1}(0)=2$；$w_{s2}(0)=4$。在对系统施加慢时变扰动 $d_1=c_1\sin(0.0314k)$ 和随机噪声干扰 $d_2=c_2\mathrm{randn}(M,1)$ 的情况下，试求系统输出 y，试用 MATLAB 语言编写系统计算机仿真程序，并打印出仿真结果。

第 8 章　非线性动态系统的其他辨识方法

非线性系统是广泛存在的，严格地讲，几乎所有的实际系统都是非线性系统。在工程中，很多非线性系统都可以用线性系统足够好地近似，但对非线性程度严重的系统必须采用特殊的方法来处理。除了前面讨论的非线性辨识方法以外，还有一些其他的处理方法。本章主要讨论非线性离散时间动态系统的辨识问题。关于非线性动态系统的辨识问题，一般可以通过最优化方法来解决，读者可参阅有关书籍，这里不再赘述。

非线性系统类型很多，不可能由一种辨识理论来解决所有非线性系统辨识问题，因此必须针对不同的非线性系统分别来研究其辨识问题。

不同类型的非线性系统可以用不同的模型来描述，其中最重要的一类模型是非线性微分方程或差分方程模型。这类模型的特点是：可以通过物理规律直接地或经过某些近似后自然地得到模型；在数值上容易处理；符合节省参数的原则。其他一些重要的已研究的非线性模型是：

(1) 用 Volterra 级数展开表示的多项式系统；

(2) 含有无记忆非线性的 Hammerstein 模型；

(3) 双线性系统模型；

(4) 解析-线性(analytic-linear)系统模型；

(5) $\dot{x}=f(x,c)$，$x\in\mathbf{R}^m$ 型的自治非线性模型等。

另外，还有一些特殊类型的非线性系统模型，如继电式、滞环、极限圈及混沌运动等。特别指出，用于系统辨识的神经网络加上训练后的权值和阈值参数也可以表示被辨识系统的模型。

目前，针对不同类型的非线性模型已提出了不少辨识方法。由于非线性系统的复杂性，用于辨识的数学工具也是多种多样的，有的还相当深奥。限于篇幅这里只能介绍两种比较典型的非典型模型的辨识方法。

8.1　Volterra 级数的表示及其辨识方法

用多阶脉冲响应表示的 Volterra 级数可对一类广泛的非线性过程给出相当一般的非参数表达式。下面推导非线性系统 Volterra 级数的表达式[72]。

8.1.1 非线性系统 Volterra 级数的表示

考虑有记忆的非线性过程。假定,当 $t<0$ 时,输入信号 $u(t)=0$,此外还假定这个记忆是有限的,亦即系统以前的输入对目前的输出是有影响的,但是当 τ 足够大时,$u(t-\tau)$就不再对响应 $y(t)$产生影响了。这些假定自然满足过程的稳定性和物理可实现性(但是,一个理想的积分器是有无限记忆的)。过程输入量可以通过具有有限面积的矩形脉冲来近似,如图 8.1 所示。以时间间隔 Δt 为周期,对 $\tau\leqslant t_0$ 时,从$u(t)$上得到的采样值分别记为 u_1、u_2、$\cdots$、u_N。N 应该选取得足够大,使当 $n>N$ 时,u_N将不再影响$y(t_0)$,于是响应 $y(t_0)$便被近似为 N 个变量的函数 $f(u_1,u_2,\cdots,u_N)$。设 $y(0)=f(0,0,\cdots)=0$,利用多维泰勒级数在零点处展开,我们得到(对某一时刻 t)系统的输出为

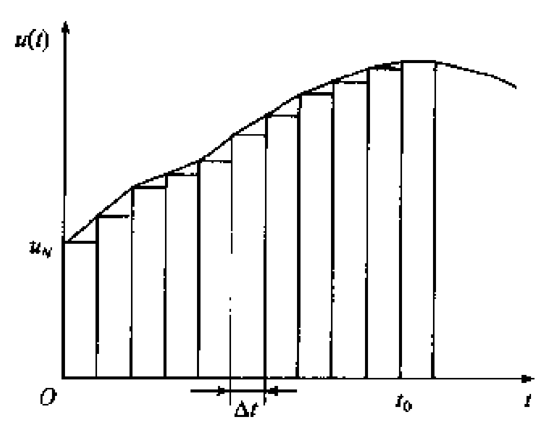


图 8.1 输入量用矩形脉冲表示图

$$
\begin{aligned}
\tilde{y}(t) &= (a_1u_1 + a_2u_2 + \cdots + a_Nu_N) \\
&\quad + (a_{11}u_1^2 + a_{12}u_1u_2 + \cdots + a_{NN}u_N^2) \\
&\quad + (a_{111}u_1^3 + a_{123}u_1u_2u_3 + \cdots + a_{NNN}u_N^3) + \cdots \\
&= \sum_{i=1}^{N} a_i\, u_i + \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}u_i\, u_j + \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} a_{ijk}u_iu_ju_k + \cdots \\
&= \tilde{y}_{\text{lin}} + \tilde{y}_{\text{quadr}} + \tilde{y}_{\text{cub}} + \cdots
\end{aligned} \tag{8.1}
$$

式(8.1)称为 Kolmogorov-Gabor 多项式,式中,$\tilde{y}$表示 y 的近似值;$\tilde{y}_{\text{lin}}$表示线性项求和,$\tilde{y}_{\text{quadr}}$,$\tilde{y}_{\text{cub}}$分别表示二次项求和及三次项求和。在前面已经指定的条件下,如果 Δt 比系统时间常数小,则$\tilde{y}(t)$和 $y(t)$非常近似。

若使所有的 a_{ij}和a_{ijk}为零,则仅剩 $\tilde{y}_{\text{lin}}=\sum_{i=1}^{N} a_iu_i$,这是对线性过程的卷积分的近似。

这可做如下说明:

令 $a_i=h_i\Delta t$,其中 h_i 是脉冲高度,Δt 是脉冲宽度,因此

$$
\tilde{y}_{\text{lin}}(t_0) = h_1u_1\Delta t + h_2u_2\Delta t + \cdots - \sum_{i=1}^{N} h_iu_i\Delta t \tag{8.2}
$$

考虑到图 8.1 中 u_i 的编号方向,当 $\Delta t\to 0$, $N\to\infty$ 以及 $N\Delta t=t_0$ 时,此式变为

$$\tilde{y}_{\text{lin}}(t_0)=\int_0^{t_0}h_1(\tau)u(t_0-\tau)\mathrm{d}\tau \tag{8.3}$$

这里 $h_1(\tau)$ 表示系统的一阶响应(核),相当于线性系统的脉冲响应。若使两个脉冲加到系统上,那么对于线性和的作用遵循叠加原理

$$h_iu_i\Delta t+h_ju_j\Delta t \tag{8.4}$$

但在平方和、立方和等和式中,脉冲间的相互作用同样是明显存在的。假定,$a_{ij}=h_{ij}\cdot\Delta t\Delta t$,则对 $\tilde{y}_{\text{quadr}}$ 的作用是

$$h_{ii}u_i^2\Delta t\Delta t+2h_{ij}u_iu_j\Delta t\Delta t+h_{jj}u_j^2\Delta t\Delta t \tag{8.5}$$

对 $\tilde{y}_{\text{quadr}}$ 取极限则变成

$$\tilde{y}_{\text{quadr}}(t_0)=\int_0^{t_0}\int_0^{t_0}h_2(\tau_1,\tau_2)u(t_0-\tau_1)u(t_0-\tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2 \tag{8.6}$$

在这种情况下

$$\tilde{y}(t)=\tilde{y}_{\text{lin}}(t)+\tilde{y}_{\text{quadr}}(t)+\tilde{y}_{\text{cub}}(t)+\cdots \tag{8.7}$$

就变为 Volterra 级数

$$\begin{aligned}y(t)=&\int_0^t h_1(\tau)u(t-\tau)\mathrm{d}\tau\\&+\int_0^t\int_0^t h_2(\tau_1,\tau_2)u(t-\tau_1)u(t-\tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2\\&+\int_0^t\int_0^t\int_0^t h_3(\tau_1,\tau_2,\tau_3)u(t-\tau_1)u(t-\tau_2)u(t-\tau_3)\mathrm{d}\tau_1\mathrm{d}\tau_2\mathrm{d}\tau_3\end{aligned} \tag{8.8}$$

所以用 Volterra 级数来描述非线性过程,是用卷积分描述线性过程的直接推广,线性系统的脉冲响应 $h(\tau)$ 被各阶脉冲响应(核)$h_1(\tau),h_2(\tau_1,\tau_2),h_3(\tau_1,\tau_2,\tau_3)$ 等所取代。这里称 h_i 为第 i 阶脉冲响应,称积分

$$\int_0^t\int_0^t\cdots\int_0^t h_i(\tau_1,\tau_2,\cdots,\tau_i)u(t-\tau_1)\cdots u(t-\tau_i)\mathrm{d}\tau_1\cdots\mathrm{d}\tau_i \tag{8.9}$$

为第 i 次泛函。

8.1.2 Volterra 级数的辨识

假定非线性系统是稳定的,并具备有限的建立时间,亦即该系统为有限记忆的,为了辨识内核 $h_n(\tau_1,\tau_2,\cdots,\tau_n)$,需把 Volterra 级数离散化,即用它的采样数据形式近似

$$\begin{aligned}y(k)=&\sum_{i=0}^{p}h(i)u(k-i)+\sum_{i=0}^{p}\sum_{j=0}^{p}h(i,j)u(k-i)u(k-j)\\&+\sum_{i=0}^{p}\sum_{j=0}^{p}\sum_{m=0}^{p}h(i,j,m)u(k-i)u(k-j)u(k-m)+\cdots\end{aligned} \tag{8.10}$$

式中,$k\geqslant p$;采样周期为 T;$t_k=kT$;pT 是建立时间;$h(i)=h_1(\tau=iT)T$,

$h(i,j)=h_2(\tau_1=iT,\tau_2=jT)T^2$,等等,为了简化起见,在式(8.10)中省略了 T。

现在的目标就是从输入输出数据序列 $u(k)$ 和 $y(k)$ 中估计出 $h(i)$、$h(i,j)$ 等。这种估计可以很容易地用参考文献[72]第 4 章中的分阶段最小二乘估计来得到。如果取 Volterra 级数的前两项来近似,并假定系统输出端受干扰,即如图 8.2 所示,则

图 8.2 系统输出端受干扰图

$$z(k)=\boldsymbol{h}^{\mathrm{T}}(k)\boldsymbol{\theta}+v(k) \tag{8.11}$$

式中,$v(k)$ 是附加的随机噪声;

$$\boldsymbol{h}^{\mathrm{T}}(k)=[u(k)\cdots u(k-p)\ \vdots\ u^2(k)u(k)u(k-1)\cdots u^2(k-p)] \tag{8.12}$$

$$\boldsymbol{\theta}^{\mathrm{T}}=[h(0)\cdots h(p)\ \vdots\ h(0,0)h(0,1)\cdots h(p,p)] \tag{8.13}$$

令 $J=\sum v^2(k)$ 趋于最小,则从式(8.11)可得 $\boldsymbol{\theta}$ 的最小二乘估计。J 的求和式中 $k=1,2,\cdots,N$, $N>\dim\boldsymbol{\theta}$。关于 $\boldsymbol{\theta}$ 的详细解法可参阅文献[71]第 6 章非线性控制系统的频域分析,先找到 Volterra 级数和广义频率响应(generalized frequency response function,GFRF)的对应关系,然后用非线性系统的 GFRF 递推算法求 Volterra 的内核 $h(\tau)$(脉冲响应);对多输入多输出(MIMO)系统,可采用 MIMO 非线性系统的 GFRFM 递推算法求矩阵形式的 Volterra 的内核 $\boldsymbol{H}(\tau)$,即求出了多变量系统的脉冲响应。

8.2 复杂系统的混沌现象及其辨识

混沌是非线性系统的通有行为,控制过程出现混沌现象是不可避免的。而且控制系统中经常含有不确定的因素,因此非线性控制系统比非线性自治系统更为复杂,很容易出现混沌现象。控制系统的混沌运动来自三个方面:一是非线性控制系统本身的混沌。二是控制算法带来的混沌;三是系统离散化引起的混沌。若系统模型用状态方程来描述,对定常系统而言,只有三阶以上的系统能够产生混沌。而对于时变系统能够产生混沌至少需二阶;但一阶时间延迟系统就可能有复杂运动行为。

文献[71]的第 5 章指出:混沌的控制完全不同于常用的控制问题,它有两层意思:一是确实消除了混沌,使系统稳定到期望点或期望的轨道;另一层含义是抑制混沌,将其幅值减少到期望的范围之内。混沌的同步属于混沌的控制范围,它是在给定的相空间锁定相轨迹,即混沌的同步是控制的轨迹跟踪期望轨迹的一类特殊控制问题。

8.2.1 反馈系统和优化过程中的混沌现象

1. 反馈控制引起的混沌

考虑具有反馈延迟的离散控制系统

$$x_{k+1} = f(x_k) + u_k, u_k = u(x_{k-\tau}) \tag{8.14}$$

其相应的闭环系统为

$$x_{k+1} = f(x_k) + u(x_{k-\tau}) = F(x_k, x_{k-\tau}) \tag{8.15}$$

延迟系统比相应的无延迟系统具有更丰富的动态特性。文献[71]通过研究一类二次映射来加以说明。考虑

$$f(x_k) = 1 - ax_k^2,\ u(x_{k-\tau}) = bx_{k-\tau} \tag{8.16}$$

取 $a=1.4, b=0.3$,则

$$\tau \neq 0: x_{k+1} = 1 - 1.4x_k^2 + u_k, u_k = 0.3x_{k-\tau} \tag{8.17}$$

$$\tau = 0: x_{k+1} = 1 - 1.4x_k^2 + 0.3x_k \triangleq F(k) \tag{8.18}$$

先研究无延迟系统式(8.18)的动态特性。容易求得它有两个不动点 $x_1=0.63$,$x_2=-1.13$,且均不稳定(排斥子),故系统式(8.18)不会收敛于一点。

定理 8.1 系统式(8.18)当初值 $x<x_1$ 或 $x>x'$ 时发散,其中 $x'=1.77$。

可以证明,系统式(8.18)具有不变区间 $I=[x_1, x']$,在 I 上非拓扑传递。

利用大量仿真数据可算得系统式(8.18)在 I 内的 Lyapunov 指数 $\sigma \leqslant 0$,故它不可能呈混沌态。但它在 I 内不收敛于一点,又不发散,故只能是周期或拟圆周运动。

再考虑延迟系统式(8.17)。取 $\tau=1$ 时便得到著名的 Henon 映射,而 Henon 映射是混沌的,可见,反馈延迟引起了混沌动态行为。

下面讨论连续非线性闭环系统

$$\begin{cases} \dot{x} = f(x) + u \\ u = g(x(t-\tau)) \end{cases} \tag{8.19}$$

式(8.19)可改写为

$$\dot{x} = f(x) + g(x(t-\tau)) \triangleq F(x, x_\tau) \tag{8.20}$$

式中,$x_\tau = x(t-\tau)$。设 x^* 是式(8.20)的一个平衡点,在 x^* 附近展成泰勒级数并保留到一次项,可得

$$\dot{z} = az + bz_\tau \tag{8.21}$$

式中,$z = x - x^*$,$a = \left.\dfrac{\partial F}{\partial x}\right|_{x=x^*}$,$b = \left.\dfrac{\partial F}{\partial x_\tau}\right|_{x_\tau=x^*}$。则式(8.20)在 x^* 附近的局部稳定性可由研究式(8.21)得到,对此有如下所述主要的结果:

定理 8.2　设式(8.21)满足:① $|b|<|a|$,或② $|b|>|a|$, $\tau<\arccos(-a/b)\cdot\sqrt{b^2-a^2}$ 时,则 $\mathrm{Re}(\lambda)<0$,此时,x^* 即为式(8.20)的一个局部渐近稳定平衡点。

若有 $|b|<|a|$,则由该定理立即得出,当 $\tau\geqslant\arccos(-a/b)\sqrt{b^2-a^2}$ 时,平衡点将失稳,从而使得式(8.20)的动力学变得复杂起来。随着延迟时间的增加,系统将表现出从振荡直到混沌等一系列复杂的动态。因此,对于任何实际的反馈控制系统,当系统中存在着明显的非线性时,设计控制算法必须考虑完成该算法所需的时间,它应使系统总的延迟时间不超过上述定理所界定的范围。

2. 动态系统优化引起的混沌

考虑一类带参数 a 的非线性动态系统

$$x_{k+1}=F(a,x_k),k=0,1,2,\cdots \tag{8.22}$$

式中,$F:\mathbf{R}\to\mathbf{R}$,$a\in\Omega$ 为待优化的参数,Ω 是容许参数集合。对系统式(8.22)的参数 a 进行优化,即求出某一 $a=a^*\in\Omega$ 使目标函数 J 最优。比如定义 J 为

$$\max_a J=\sum_{k=0}^{N-1}f(a,x_k) \tag{8.23}$$

该优化问题的直接解析求解可利用拉格朗日乘子法和离散变分法,也可进行数值计算。讨论优化问题时常把注意力集中于优化目标上而忽视系统的动态行为。当

$$x_{k+1}=F(a^*,x_k),\quad k=0,1,2,\cdots \tag{8.24}$$

这是一维迭代映射,其稳定性和动态行为可应用文献[71]的第 2 和第 3 章的中心流形定理及结构稳定性的方法加以处理。不同的参数 a 往往得出不同的目标值,也可能导致完全不同的动态行为,甚至出现混沌。下面举两例说明之。

考虑典型系统

$$x_{k+1}=F(a,x_k)=ax_k(1-x_k),x_0=0.4,A=[1,4] \tag{8.25}$$

取 $\max\limits_a J_1=\sum\limits_{k=0}^{100}x_k$ 时,$a^*\approx3.709$,对应的 $\max\limits_a J_1=67.777$。若取 $\max\limits_a J_1=\sum\limits_{k=0}^{100}x_k^2$ 进行优化,也能得到几乎同样的 a^*,在 A 内($J_i\sim a,i=1,2$)的仿真表明,在 $a=a^*$ 时,$i=1,2$ 两种情况均出现混沌。

3. 采样控制系统的分岔与混沌

有许多原因导致连续系统的离散化,如计算机控制、数值计算等。用离散化技术处理连续系统时离散步长或周期是一个重要参数,它对信号恢复和系统稳定性等都有一定影响。这里从动态系统理论的角度来研究这一问题。

考虑一维非线性反馈控制系统

$$\dot{x} = f(x) + u(t), \quad u(t) = u(x) \tag{8.26}$$

其闭环控制系统为

$$\dot{x} = f(x) + u(x) \triangleq G(x) \tag{8.27}$$

可采用多种方法将连续系统式(8.27)离散化,一种简单的前向差分法表示为

$$\dot{x} \approx (x_{k-1} - x_k)/T, \quad x(t) = x_k \tag{8.28}$$

式中,T 为采样周期。于是,闭环系统式(8.27)近似为

$$x_{k-1} = x_k + TG(x_k) \triangleq F(x_k, T), \quad k = 1,2,\cdots \tag{8.29}$$

容易证明,系统式(8.27)的平衡点 x^e 即为相应离散系统式(8.29)的不动点 x^f,并且:若 x^e 不稳定,则 x^f 不稳定;若 x^f 稳定,则 x^e 稳定。随着采样周期 T 的变化,离散系统式(8.29)呈现出比连续系统式(8.27)更为丰富的动态特性,如奇异吸引子和混沌等。

定理 8.3　当系统(8.29)的不动点 x^f 满足 $G'(x^f)<0$ 时,若 $0<T<-2/G'(x^f)$,则 x^f 稳定;若 $T>-2/G'(x^f)$,则 x^e 不稳定。

证明　因 $G'(x^f)<0$,故当 $0<T<-2/G'(x^f)$ 或 $T>-2/G'(x^f)$ 时,$|F'(x^f)| = |1+TG'(x^f)|$ 分别为 <1 和 >1,于是由稳定性定义,知 x^e 分别是稳定的和不稳定的。

定理 8.4　当系统(8.29)的不动点 x^f 满足 $G'(x^f)<0$ 时,若 T 由小到大穿越 $T=-2/G'(x^f)$,则系统产生分岔。

下面通过一个实例,分析采样周期 T 变化引起的分岔和混沌的机理,设

$$f(x)=ax^3-bx, u(x)=-c(x), \quad a>0, \quad b>0, \quad c>0 \tag{8.30}$$

进而设反馈系数 $c=a-b$,则式(8.27)和式(8.29)分别写成

$$\dot{x} = a(x^3 - x) = G(x) \tag{8.31}$$

$$x_{k+1} = x_k + T[a(x_k^3 - x_k)] = F(x_k, T) \tag{8.32}$$

系统式(8.31)有三个平衡点 $x_1^e=0, x_{2,3}^e=\pm 1$,且 x_1^e 稳定而 $x_{2,3}^e$ 不稳定。而式(8.32)有三个不动点 $x_i^f=x_i^e(i=1,2,3)$,并且 $x_{2,3}^f$ 不稳定。若采样周期 T 由小到大穿越 $T=-2/a$,则 x_1^f 由稳定变为不稳定(定理 8.3),产生分岔(定理 8.4)。

定理 8.5　当采样周期 $T=4/a$ 时,离散控制系统式(8.32)在不变区间 $I=[-1,1]$ 内呈混沌状态,证明见文献[71]的 5.2 节。

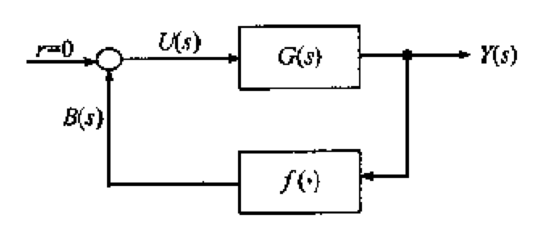


图 8.3　非线性反馈控制系统

8.2.2　基于控制理论的混沌分析方法

传统的控制理论也可以用来处理控制系统中的混沌现象,下面介绍如何利用描述函数分析混沌。

对于如图 8.3 所示的一类典型的非线性

控制系统为

$$G(s)=\frac{k(s^{n-1}+b_{n-1}s^{n-2}+\cdots+b_2s+b_1)}{s^n+a_ns^{n-1}+\cdots+a_2s+a_1} \tag{8.33}$$

式中，k 为大于零的可调增益。系统的平衡点可由下述方程得到

$$y+G(0)f(y)=0 \tag{8.34}$$

设它的解是 $y=E_i, i=1,2,\cdots$。现设 $r=0$，做如下假设：

假设 8.1　系统输出形式为 $y_0(t)=A+B\sin\omega t$ $(B>0,\omega>0)$。

假设 8.2　非线性输出，即 $f[y_0(t)]$ 可由傅里叶级数展开成

$$f(y(t))=F_0(A,B,\omega)+\sum_{k=1}^{\infty}[F_{KR}(A,B,\omega)\sin k\omega t+F_{KI}(A,B,\omega)\cos k\omega t] \tag{8.35}$$

式中，F_{KR}，F_{KI} 是相应的谐波系数。忽略式(8.35)中的高次谐波，得

$$N_0=F_0/A,\quad N_1=(F_{1R}+\mathrm{j}F_{1I})/B$$

$$N_0(A,B)=\frac{1}{2\pi A}\int_{-\pi}^{\pi}f(A+B\sin\varphi)\mathrm{d}\varphi \tag{8.36}$$

$$N_1(A,B)=\frac{1}{\pi B}\int_{-\pi}^{\pi}f(A+B\sin\varphi)\mathrm{d}\varphi \tag{8.37}$$

分析表明，当系统的极限环和具有稳定特征的平衡点产生交互作用，且系统具有一定的滤波作用时，反馈系统会产生混沌运动。

定理 8.6　Lur'e 非线性反馈系统当其极限环和平衡点相互作用时，系统能够产生混沌的近似必要条件为

$$B\geqslant|E-A| \tag{8.38}$$

对实际的 Lur'e 非线性反馈系统，可按下列步骤分析其混沌现象：

(1) 解 $A[1+G(0)N_0(A,B)]=0$ 和 $1+G(\mathrm{j}\omega)N_1(A,B)=0$ 可得极限环 $y=y_0(t)$ 存在的条件；

(2) 解方程 $y+G(0)f(y)=0$ 得系统的平衡点 $y=E_i, i=1,2,\cdots$；

(3) 由劳斯判据和奈奎斯特稳定性判据可获得平衡点和极限环的稳定性；

(4) 由 $y_0(t)=A+B\sin\omega t$，$B\geqslant|E-A|$，即可得到极限环和平衡点相互作用的条件。

若图 8.3 中非线性反馈环节取为 $f[y(t)]=y^3(t)/3$，前向通路环节为

$$G(s)=\frac{k(s^2+b_2s+b_1)}{s^3+a_3s^2+a_2s+a_1},\quad k>0$$

可解得系统的输出平衡点为

$$E_1=-\sqrt{-3a_1/kb_1},\quad E_2=0,\quad E_3=-\sqrt{-3a_1/kb_1},\quad k>0,\quad a_1b_1<0$$

由于它为三阶系统，从而可能出现复杂运动行为。此时系统的闭环极点具有以下特征：实部有正有负；无零实部极点(即虚轴上无极点)；所有极点之和小于零；存在复共轭极点。这些保证了系统是耗散的，具有有界、总体吸引、局部排斥等特性。这时系统在稳定的流形上能收缩，在不稳定的流形上可以扩张。现考虑直流分量为零的情况，应用上述步骤对该系统进行分析，可得出下列结论：

定理 8.7 对图 8.3 所示非线性系统，若

$$G(s) = \frac{k(s^2 + b_2 s + b_1)}{s^3 + a_3 s^2 + a_2 s + a_1}, \quad k > 0$$

$$f[y(t)] = y^3(t)/3$$

则当参数 $a_i(i=1,2,3)$，$b_i(i=1,2)$同时满足条件 1 和条件 2 时，

条件 1：$(a_2 + a_3 b_2 - b_1)^2 > 4b_2(a_3 a_2 - a_1)$ (8.39)

条件 2：
$$\begin{aligned}&(1)\, b_1>0, \Delta > a_2 + a_3 b_2 - b_1;\\ &(2)\, b_2>0, \Delta < a_2 + a_3 b_2 - b_1;\\ &(3)\, b_1<0, \Delta < -(a_2 + a_3 b_2 - b_1);\\ &(4)\, b_2<0, \Delta > -(a_2 + a_3 b_2 - b_1);\end{aligned} \tag{8.40}$$

式中，$\Delta = \sqrt{(a_2 + a_3 b_2 - b_1)^2 - 4b_2(a_3 a_2 - a_1)}$，则系统随 k 变化会出现混沌运动。

例 8.1 考虑系统 $G(s) = \dfrac{k(s^2+3s+66.25)}{s^3+0.5s^2+7s-15}$，其零点为 $z_1 = -1.5 - \mathrm{j}8$，$z_2 = -1.5 + \mathrm{j}8$，极点为 $S_1 = -1 - \mathrm{j}3$，$S_2 = -1 + \mathrm{j}3$，$S_3 = 1.5$。极点分布具有上面的四个特征，系数 $a_i(i=1,2,3)$，$b_i(i=1,2)$满足定理 8.7，故系统参数 k 的变化会出现混沌运动。

8.2.3 混沌识别与混沌系统辨识

1. 未知模型时混沌系统的识别

对于大多数情况，人们并不了解系统的精确数学模型，甚至连系统的阶次也不知道，只能得到一组输入输出数据，即系统某些状态的时间序列。而且这种未知结构系统往往具有不可测量的噪声。这就是第 1 章所述的“黑箱”辨识问题，即具有不可测输入激励的未知结构的动力系统。对这种“黑箱”系统，首先必须解决的是如何区分混沌与噪声，即对混沌姿态进行识别。

对于时间序列中的噪声，一般来说分为两种，动力学噪声和观测噪声，这两种噪声对系统的分析都有重大的影响。使用计算得到的关联维数可以区分混沌与噪

声。对于混沌运动,随着嵌入维数 d 的增大,关联维数 D 将收敛于某一确定值;而对于噪声,关联维数将随着嵌入维数的增大而增大,如图 8.4 嵌入维数 d 与关联维数 D 的关系。但有时用关联维数来分析系统的混沌非线性性质是失效的。

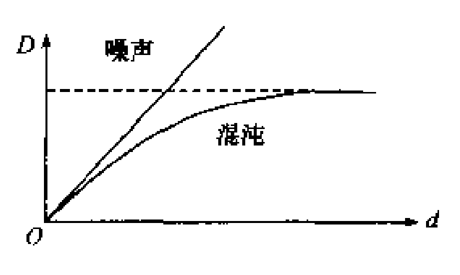


图 8.4 嵌入维数与关联维数的关系

对于未知模型混沌系统的具体识别方法主要有三种:序列的可预测性方法、非线性预报方法和替代数据法。

1) 序列的可预测性方法

由于混沌是"无序中的有序",混沌时间序列是短期可预测而长期不可预测的,噪声是不可预测的,故由序列的可预测特性来区分混沌与噪声。对于一个时间序列来说,首先我们要进行相对空间重构,在此基础上分析序列的可预测特性,具体由原始序列和预测序列的相关系数

$$r=\frac{\sum_{n=1}^{p}[x(n)-\bar{x}][\hat{x}(n)-\hat{x}]}{\sqrt{\sum_{n=1}^{p}[x(n)-x\bar{x}]^2}\sqrt{\sum_{n=1}^{p}[\hat{x}(n)-\bar{\hat{x}}]^2}} \tag{8.41}$$

来度量。式中,$x(n)$,$\hat{x}(n)$分别为序列的实际值和预测值;$\bar{x}$,$\bar{\hat{x}}$分别为相应的均值;p 为预测总的点数。嵌入维数和预测步数都对 r 有影响。

改变嵌入维数 d 对序列作单步预测时,对于混沌时间序列而言,开始阶段 r 达到最大值。随后,r 随嵌入维数的增加而减少。对于噪声时间序列而言,r 几乎不随嵌入维数改变。

取 r 最大时的嵌入维数 d 值对序列作多步预测。对于混沌序列而言,r 随预测步数的增加而急剧下降;而随机噪声序列则几乎为零,不改变。

2) 非线性预报方法

1990 年,Sufihara 和 May 提出了一种具有动态分析实验数据特点的非线性预报方法。其基本思想是混沌运动虽然对初值敏感,长期不可预测,但由于它又具有确定性运动的特征,短期可以预测,这样就可以较好地区分混沌和噪声。

依据非线性预报法的原理,下面给出这种用于区分混沌与噪声方法的实现算法。

算法 8.1 非线性预报方法

① 先将序列$\{u(k)\}$分为两半,前一半 $N/2$ 个数用作预报,后一半数则用于比较预报与实际序列值的差别。

② 取 $\Delta u(k)=u(k+1)-u(k)$,$k=1,2,\cdots,N/2$。

③ 选定嵌入维数 d 和时间延迟 τ，并形成 d 个数的数列 $(\Delta u(k), \Delta u(k+\tau), \cdots, \Delta u(k+(d-1)\tau))$，它表示 d 维空间的一个点 $\boldsymbol{x}(k)$。

④ 在 $\boldsymbol{x}(k)$ 所有的邻点中，按照与 $\boldsymbol{x}(k)$ 距离最小的原则，选出 $d+1$ 个 d 维的邻点 $\boldsymbol{x}(i)$，构成包含 $\boldsymbol{x}(k)$ 点的最小单纯形。

⑤ 通过上面的邻点 $\boldsymbol{x}(i)(i=1,2,\cdots,d+1)$ 和时间 $T_p = p\tau$（p 为预报的步数）后的演化点 $\boldsymbol{x}^p(i)$，以各邻点与其 p 步预测后的点 $\boldsymbol{x}(i)$ 距离的指数作为权重，按下式确定

$$\overline{\boldsymbol{x}^p(i)} = \frac{1}{i-1}\sum_{i=1}^{d+1}\exp[-\|\boldsymbol{x}(k)-\boldsymbol{x}(i)\|]\boldsymbol{x}^p(i) \qquad (8.42)$$

⑥ 以 $\boldsymbol{x}(k)$ 点 p 步后的实际值为横坐标，$\boldsymbol{x}(k)$ 点 p 步后的预报值 $\overline{\boldsymbol{x}^p(k)}$ 为纵坐标，作对应的坐标图。如果点聚集在两坐标的对角线附近，则表示预报准确，时间序列可以预测，即此序列是混沌的，否则，说明该序列是随机序列。

3) 替代数据法

Oshome 和 Theiler 等基于随机系统中零假设的概念，提出了一种利用替代数据，即直接由原始时间序列产生随机数据的统计量，来判定该序列非线性的特性的方法。这种方法是目前区分混沌与噪声的比较好的方法，在实际对象中非常实用。

替代数据法是一种统计检验方法，它的基本思想是假设所研究的时间序列为具有某种特性的非线性随机过程，如非线性高斯过程，然后用特定的方法对原始数据进行随机化，并根据原假设保留数据的某些非线性特性，如均值、方差、频谱等，这就是所谓的替代数据。随后对原始数据和替代数据分别计算某些检验参量，再用数理统计方法来确定两者之间是否有显著差异。如果有，则假设不成立，说明研究对象不是所假设的非线性随机系统。

为了检验原始数据和替代数据是否存在显著差异，需要计算某些非线性特性的统计参量。目前常用的检验参量有关联维数、李雅普诺夫、非线性预测误差、局部流等方法。

替代数据法一般的研究步骤为：

① 需要确定原假设 H_0，一般假设原序列为一非线性随机过程；

② 对原序列随机化，产生所需要的替代数据，并使替代数据保留原序列在原假设 H_0 下的非线性特性；

③ 选择非线性检验参量；

④ 分别计算原始数据和替代数据的检验参量，并用数理统计方法来确定原假设 H_0 是否成立。

目前常用的替代数据法有三种：随机化相位替代数据、混序傅里叶变换替代数据和高斯标度替代数据。随机化相位替代数据对原序列相位加入具有某种特性的均匀分布的随机变量；混序傅里叶变换替代则对随机化行为替代数据保持幅值不

变,而重新进行升序或降序排列处理;而高斯标度替代数据则对原序列相位加入高斯分布的随机序列,再产生随机化相位高斯过程(有色噪声)。依据随机化相位替代数据法的原理,下面给出这种用于区分混沌与噪声方法的实现算法。

设$\{u(k)\}(k=1,2,\cdots,N)$为从试验中获得的一组反映系统状态的时间序列。

算法 8.2　替代数据法

① 对时间序列

$$u(k)=u(k,\Delta k),\quad k=0,1,2,\cdots,N-1$$

作傅里叶变换

$$U(\omega)=\frac{1}{2\pi}\sum_{k=1}^{N-1}u(k,\Delta t)\mathrm{e}^{\mathrm{j}k\omega\Delta t} \tag{8.43}$$

或

$$U(\omega)=A(\omega)\mathrm{e}^{\mathrm{j}\varphi(\omega)},\quad \omega=0,\pm\Delta\omega,\cdots,\pm N\Delta\omega,\quad \Delta\omega=\frac{1}{N\Delta t} \tag{8.44}$$

② 对频率变换$U(\omega)$增加一个均匀分布在区间$(0,2\pi)$的随机量$\Psi(\omega)$,并使$\Psi(\omega)=\varphi(-\omega)$,则有

$$\overline{U(\omega)}=A(\omega)\,\mathrm{e}^{\mathrm{j}[\varphi(\omega)+\Psi(\omega)]} \tag{8.45}$$

③ 对$\overline{U(\omega)}$进行傅里叶反变换,即可得替代时间序列

$$u(k)=\frac{1}{N}\sum_{i=0}^{N-1}\overline{U(\omega)}\,\mathrm{e}^{-k(2\pi\mathrm{j}k/N)} \tag{8.46}$$

④ 进行显著性检验

$$\eta=\frac{|Q_0-uH|}{\sigma H} \tag{8.47}$$

式中,Q_0为原始序列的一个判别统计量,uH和σH分别是替代数据对应的统计量Q_0的均值和方差。

⑤ 判别:如果步骤④中的η大,则表示替代数据和原始数据有显著差异,即说明有非线性序列存在,否则,说明该时间序列是噪声。

2. 混沌系统的辨识

混沌的辨识是指确定一个混沌系统的数学模型,包括结构辨识和参数辨识两部分。为了能从时间序列中得到动力系统相空间的几何机构,Packard 等人做了创造性工作,他们把一维时间序列嵌入到d维空间中,采用时间延迟技术重构相空间,Takens 和 Mane 证明了只要$d>2D+1$,其中D为吸引子的分维数。则映射

$$\Psi:\mathbf{R}^n\to\mathbf{R}^m,\ \mathbf{R}^n\text{ 为相空间},\ \mathbf{R}^m\text{ 为嵌入空间}$$

是在吸引子附近的光滑、一对一映射,从而嵌入空间中吸引子的几何特性与原动力系统中的几何特性等价。在 1985 年,Eekman 等人证明了只要$d>D$,嵌入空间中

点集的维数就等价于吸引子的维数,从而解决了相空间重构理论中的一个重要问题。理论上讲对于具有无穷精度的无限长时间序列,延迟参数 τ 的选取是可以任意的,d 只需要大于 D。而在实际应用中,$(d-1)\tau$ 中 d 和 τ 的选取十分重要。若 τ 太小,嵌入空间中相点矢量各分量的冗余度增大;太大会造成分维估计值偏大,则使相点矢量各分量近似相等,在相空间中吸引分子就会为一条对角直线。Broomhed 和 King 提出了一种主分量分析法,即先选定 $d-1$ 值,增加 d 的同时,减少 τ,但保持$(d-1)\tau$ 不变。另外还可用线性不相关时间、交互信息时间和拓扑原则来重构相空间。相空间重构的方法比较成熟,但只是在某种意义下等价重构出原系统一些重要的动力学特性,并非与原系统完全等价。

至于对于混沌系统如何获得其精确的数学模型,进行具体的参数辨识,是一个有待于深入研究的问题,目前报道这方面的文献很少。Qammer, Aruirre 等在这方面做了一定的工作。他们所得到的辨识方法和传统的控制理论中的辨识不同,但由于这些方法计算量大,很难用于在线控制。Ljung 阐明可以用传统控制理论中的辨识方法如最小二乘法、递推最小方差法来确定混沌系统的模型。采用这种方法一般需先选择一类模型,然后对这类模型利用模型检验和代价最小的方法进行适当选择,不具有普遍性。

8.3 小　　结

本章介绍了复杂非线性动态系统辨识的两种方法。一种是 Volterra 级数的表示及其辨识方法,先找到 Volterra 级数和广义频率响应(generalized frequency response function, GFRF)的对应关系,然后用非线性系统的 GFRF 递推算法求 Volterra 的内核 $h(\tau)$,即系统的脉冲响应;另一种是复杂系统的混沌现象及其辨识,讨论了三种对于未知模型混沌系统的具体识别方法:即序列的可预测性方法、非线性预报方法和替代数据法;只有能识别混沌,才能进而控制混沌或诱导混沌。至于对混沌系统如何获得其精确的数学模型,进行具体的参数辨识,是一个有待于深入研究的问题。

习　　题

1. 试用 Volterra 级数来描述二阶非线性系统的脉冲响应。
2. 如何区分混沌和噪声?
3. 对于未知模型混沌系统的具体识别方法有哪几种?
4. 抑制混沌和诱导混沌的实质是什么?

参 考 文 献

[1] Billings S A. Identification of Nonlinear System-A Survey, IEE Pore. Pt. D., 1980, 5:272～285

[2] M F Abbod, D A Linkens et al. A Blackboard Software Architecture for Intelligent Control Systems, Kybernetes: The International Journal of Systems & Cybernetics, 2001, 29(8): 999～1015

[3] Liu ChunLing, La. ChiChan et al. A Neural Network for Liner Matrix Inequality Problems. IEEE Transaction on Automation Control, 2000, 45(9): 1078～1092

[4] Zhang Ying, Wan Chanyun et al. Discrete-Time Robust Backtepping Adaptive Control for Nonlinear Time-Varying Systems. IEEE Transaction on Automation Control, 2000, 45(9): 1760～1761

[5] Roy, Asim, Govil, Sandeep. A Neural-Network Learning Theory and a Polynomial Time RBF Algorithm. IEEE Transactions on Neural Networks, 1997, 8(6): 1301－1313

[6] Chen S and Billings S A. Neural Networks for Nonlinear Dynamic System Modelling and Identification N. Int. T. Control. 1992, 56: 319～346

[7] Narendra K S and Parthasarathy K. Identification and Control of Dynamic System Using Neural Network. IEEE Trans. Neural Network, 1990, 1: 4～27

[8] Burr D J. Expeinments on Neural Net Recognition of Spoken and Written Text. IEEE Trans. Accoust., Speech, Signal Processing, 1988, 36: 1162～1168

[9] 侯媛彬等. H_∞控制理论的模型匹配. 西安矿业学院学报, 1996(1)

[10] Hopfield J J. Neural Networks and Physical System with Emergent Collective Computational Abilities, Proc. Nat. Acad Sci. US, 1982, (79): 2554～2558

[11] Hopfield J J and Tank D W. Neural Computation of Decision in Optimization Problems, Biolog. Cyberm, 1985, (52): 141～152

[12] Zadeh L A. From Circuit Theory to System Theory, Proc, IRE, 1962, 50

[13] 韩光文. 辨识与参数估计. 北京: 国防工业出版社, 1980

[14] Goodwin G G and Payne R L. Dynamic System Identification—Experiment Design and Data Analysis, Academic Press, 1977

[15] Leontaritis I J and Billings S A. Input-output Parametric model for Non-linear Systems—part 1: Deterministic Non-linear Systems; Part 2: Stochastic Non-linear Systems, International Journal of Control, 1987, 41: 303～344

[16] Chen S, Cowan C F N, Billings S A et al. A Parallel Recursive Prediction Error Algorithm for Training Layered Neural Networks. International Journal of Control, 1990, 51: 1215～1228

[17] Perrone M P and Cppper L N. When Networks Disagree: Ensemble Methods for Hybird Neural Networks. Neural Networks for Speech and Image Processing, Ed. by Mammone, R. J., Chapman-Hill, London, 1993

[18] Frohlinghaus T, Wrichert A and Rujan P. Hierarchical Neural Networks for Timeserives Analysis and Control, Networks. 1994, 5: 101～116

[19] Lippmann R P. An Introduction to Computing with Neural Nets. IEEE ASSP Magazine, 1987(4): 4～22

[20] Weiland A and Leighton R. Geometric Analysis of Network Capabilities. IEEE Frist International Confer-

ence on Neural Networks, 1987,3:385~392

[21] Waibel A, Sawai H and Shikano K. Modularity and Scaling in Large Phonemic Neural Networks. IEEE Trans. A coust. ,Signal Processing, 1989,37(12)

[22] Carroll S M and Dickinson B W. Construction of Neural Nets Using Radon Transform. Proc. IJCNN, 1989,1:607~611

[23] Hiramatsu A. Traning Techniques for Neural Network Application in ATM. IEEE Communication Magz. , 1995,Oct:58~67

[24] Neves J E, Leitao M J and Almeida L B . Neural Network in B~ISDN Flow Control: TAM Traffic Prediction or Network Modeling. IEEE Communication Magz. ,1995,Oct:50~56

[25] Hunt K J and Sbabaro D. Neural Networks for Non-linear Inter-Model Control. Proc. EE Pt. D. ,1991, 138:431~439

[26] 刘乐星等. 自适应内模控制系统. 1996 年中国智能自动化学术会议论文集. 内蒙. 1996,8:501~505

[27] Jang J R. Self-Learning Fuzzy Controller Based on Temp Back Propagation. IEEE Trans. Neural Network, 1992,3(5):714~723

[28] Yamaguchi T. Self-Organizing Control using Fuzzy Neural Networks. Int. J. Control, 1992,52(2):415~439

[29] Junbong Nie and Linkens D A. Learning Control Using Fuzzy-field Self-Organizing Radial Basis Function Networks. IEEE Trans Fuzzy Syet. ,1993,1(4):281~287

[30] Narendra K S and Mukhopadhyay S. Intelligent Control using Neural Networks. IEEE Control System Magz. ,1992.12:11~18

[31] Grant E and Zhang B. A Neural Net Approach to Supervised Learning of Pole Balancing. In IEEE int. Symposium on Intelligent Control, 1989, 123~129

[32] Timo Sorsa et al. Application of Artificial Neural Networks in Process Fault Diagnosis. Automatica, 1993, 29(4)

[33] 张建华,王占林. 基于神经网络的控制系统状态监测与故障诊断方法研究. 1996 中国控制与决策学术年会论文集. 济南 1996,8:569~572

[34] Bokesenbom A S and Hood R. Genetic Algebraic Method to Control Analysis of Complex Engine Types. NAEATR~930 Washington D C, 1949

[35] Bristal E H. On a New Measure of Interaction for Multivariable Process Control. IEEE. Trans. Automatic Control, 1966, AC~11(1):133~134

[36] 徐承伟, 吕勇哉. 模糊系统的串联补偿解耦. 自动化学报, 1987,13(3):177~183

[37] Xu C W and Lu Y Z. Decoupling in Fuzzy Systems: A Cascade Compensation Approach. Fuzzy Sets and Systems, 1989, 29(2):177~185

[38] Xu C W and Lu Y Z. Decoupling Fuzzy Relational Systems—An Output Feedback Approach. IEEE Trans. Syst. Man, Cybern. , 1989, 19(2):414~417

[39] Walichiewicz L. Decomposition of Linguistic Rule in the Design of a Multidimensional Fuzzy Control Algorithm. Cybernetics Syst. Res. ,1984,2(3):185~ 193

[40] 王殿辉,柴天佑,张化光. 一种 Fuzzy 推理合成的新算法. 控制与决策, 1994,9(1):59~63

[41] Peterka V. Bayesian Approach to System Identification. In P. Eykhoff ed: Trends and Progress in System Identification. Pergamon Press, 1981

[42] 方崇智,萧德云. 过程辨识. 北京:清华大学出版社, 1988

[43] 刘宏才.系统辨识与参数估计.北京:冶金工业出版社,1996

[44] 韩崇昭,王月娟,万百五.随机系统理论 西安:西安交通大学出版社,1987

[45] 徐南荣,宋文忠,夏安邦 系统辨识.南京:东南大学出版社,1991

[46] 焦李成.神经网络系统理论.西安:西安电子科技大学出版社,1995

[47] 焦李成.非线性传递函数理论与应用.西安:西安电子科技大学出版社,1992

[48] 郭尚来.随机控制.北京:清华大学出版社,1999

[49] 冯璞刚,郭治.随机控制.北京:国防工业出版社,1998

[50] 易继锴,侯媛彬.智能控制技术.北京:北京工业大学出版社,1999

[51] 高峰.基于多ANN结构的复杂系统辨识和控制.博士学位论文.$A_3$95015.西安交通大学,1995

[52] 侯媛彬.一类非线性动态系统建模与控制研究.博士学位论文.$A_3$97339.西安交通大学,1997

[53] 侯媛彬,韩崇昭.能对非线性多变量解耦的神经网络.软件学报,1997(2).(CX9810)

[54] 侯媛彬,易继锴.降低一类神经网络灵敏度的理论和方法的研究.公路交通大学学报,1998,18(4) PR012299TC9908

[55] 侯媛彬.提高神经网络收敛速度的一种赋初值算法研究.模式识别与人工智能,2001,14(4):385~389.EX:02126894173

[56] 陈国良,王熙法等.遗传算法及其应用 北京:人民邮电出版社,1999

[57] 周明,孙树栋 遗传算法原理及应用 北京:国防工业出版社,1999

[58] Hou Yuanbin. A Decoupling Control Method with Improving Genetic Algorithm. IEEE, ICMLC02(The First International Conference on Machine Learning and Cybernetics:2112~2115, Beijing, China, Nov. 4-5,2002

[59] Hou Yuanbin, Yi Jikai. Nonlinear Identification Control and Chaos. Journal of The Beijing Polytechnic University,1997(2)

[60] Wang Mei,Hou Yuanbin, Wang Jianping. Intelligent System of Cable Fault Location and Its Data Fusion. ICMLC'02 The First International Conference on Machine Learning and Cybernetics:788~790, Beijing, China, Nov.4-5,2002

[61] 韦力,侯媛彬.非线性辨识中的模糊神经网络.公路交通大学学报,1999(3)

[62] 李秀改,侯媛彬.基于神进网络BP算法的模糊自适应控制器研究与实现.电气传动自动化,2000(4)

[63] 侯媛彬,祝海江.一种改进适应度函数的遗传神经解耦控制器.第四届全球智能控制会议,1-4:2883~2886,2002.06,ISTP收录,IDS:BV45G

[64] 侯媛彬,张建军.非线性炉群智能解耦的一种新算法.公路交通大学学报,2000,20(4)

[65] 侯媛彬,高云,汪梅.隶属函数型神经网络与模糊控制融合的解耦方法.控制与决策学术年会论文集.2001,5:539~543

[66] 侯媛彬等.带管理器的单神经元实时控制非线性系统方案的实现.信息与控制,1999,28(5)

[67] 侯媛彬.MATLAB对二维以上非线性函数学习方法研究.计算机应用研究,2002,19(增):61

[68] 解学书,钟宜生.H_∞控制理论 北京:清华大学出版社,1994

[69] 易继锴.现代控制系统设计.北京:北京工业大学出版社,1992

[70] 李人厚,秦世引.智能控制理论和方法.西安:西安交通大学出版社,1994

[71] 曹建福,韩崇昭,方洋旺.非线性系统理论及应用.西安:西安交通大学出版社,2001

[72] 刘豹,王正欧.系统辨识.北京:机械工业出版社,1993

[73] Patrick P and Snagt V D. Minimization Methods for Training Feed-forward Neural Networks. Neural Networks,1994,7(1):1~11

[74] Jacobs R A. Increased Rate of Convergence Through Learning Rate Adaptation. Neural Networks, 1988, 1:295～307
[75] Emmerson M D, Dammper R I and Hey A J D. Fault Torlance and Redundancy of Neural Nets for the Classfication of Acoustic Data. Proc. IEEE Conf. Acoustics, Speech and Signal Processing, 1992. 1053～1056
[76] 徐海银等. 于容错性网络及容错 BP 算法研究. 华中理工大学学报, 1995(2)
[77] 王德想, 李东, 孙德宝. 神经网络容错性和灵敏度的进展. 1996 中国控制与决策学术年会论文集. 1996
[78] Emmerson M D, Dammper R I. Determing and Improving the Fault Torlance of Multilaser Pereceptron in a Pattern～Recognition Application, IEEE Trans. Neural Networks. 1993, 4(5)
[79] 张贤达. 现代信号处理. 北京:清华大学出版社, 2002
[80] Isermann R. Process Identification. Springer, Berlin, 1974
[81] Landau I D. Adaptive Control -The Model Reference Adaptive Approach. Marcel Decker, New York, 1979
[82] Ljung L and T Soderstrom Theory and Practice of Recursive Identification. MIT Press, 1983
[83] Mendel J M. Identification of Decomposable Time-Varying Parameters by Means of Gradient Algorithms. Inform. Sci, 1972

目　　录